forall$\mathcal{X}$

An Introduction to Formal Logic

P.D. Magnus
University at Albany, State University of New York

fecundity.com/logic, version 1.4 [251113]

The author would like to thank the people who made this project possible. Notable among these are Cristyn Magnus, who read many early drafts; Aaron Schiller, who was an early adopter and provided considerable, helpful feedback; and Bin Kang, Craig Erb, Nathan Carter, Wes McMichael, Selva Samuel, Dave Krueger, Brandon Lee, Toan Tran, Marcus Adams, Matthew Brown, and the students of Introduction to Logic, who detected various errors in previous versions of the book.

Typesetting was carried out entirely in LaTeX2ε. The style for typesetting proofs is based on fitch.sty (v0.4) by Peter Selinger, University of Ottawa.

This copy of `forall`χ is current as of 13 de novembro de 2025. The most recent version is available on-line at `http://www.fecundity.com/logic`

Sumário

Chapter 1

O que é lógica?

A lógica é o estudo da avaliação de argumentos, separando os bons dos ruins. Na linguagem do dia a dia, às vezes usamos a palavra 'argumento' para nos referir a brigas barulhentas e cheias de hostilidade. Se você e uma amiga têm um argumento nesse sentido, as coisas não vão bem entre vocês duas.

Em lógica, não estamos interessados nesse tipo de argumento com gritos e arrancar de cabelos. Um argumento lógico é estruturado para dar a alguém uma razão para acreditar em uma certa conclusão. Eis um exemplo de argumento:

(1) Está chovendo forte.

(2) Se você não levar um guarda-chuva, vai ficar encharcado.

$\therefore$ Você deve levar um guarda-chuva.

Os três pontos na terceira linha do argumento significam 'Portanto' e indicam que a frase final é a *conclusão* do argumento. As outras frases são as *premissas* do argumento. Se você acredita nas premissas, então o argumento lhe fornece uma razão para acreditar na conclusão.

Este capítulo discute algumas noções lógicas básicas que se aplicam a argumentos em uma linguagem natural como o inglês. É importante começar com uma compreensão clara do que são argumentos e do que significa um argumento ser válido. Mais adiante, traduziremos argumentos do inglês para uma linguagem formal. Queremos que a validade formal, tal como será definida na linguagem formal, preserve pelo menos algumas das características importantes da validade em linguagem natural.

1.1 Argumentos

Quando as pessoas querem apresentar argumentos, elas frequentemente usam palavras como 'portanto' e 'porque'. Ao analisar um argumento, a primeira coisa a fazer é separar as premissas da conclusão. Palavras como essas são uma pista do que o argumento pretende ser, especialmente se — na forma como o argumento é apresentado — a conclusão vier no início ou no meio do argumento.

indicadores de premissa: já que, porque, dado que

indicadores de conclusão: portanto, logo, assim, então

Para sermos perfeitamente gerais, podemos definir um ARGUMENTO como uma sequência de sentenças. As sentenças no início da sequência são as premissas. A última sentença da sequência é a conclusão. Se as premissas forem verdadeiras e o argumento for bom, então você tem uma razão para aceitar a conclusão.

Perceba que essa definição é bastante geral. Considere este exemplo:

Há café na cafeteira.
Há um dragão tocando fagote em cima do armário.
$\therefore$ Salvador Dalí jogava pôquer.

Pode parecer estranho chamar isso de argumento, mas isso acontece porque seria um argumento péssimo. As duas premissas não têm absolutamente nada a ver com a conclusão. Ainda assim, dada a nossa definição, isso continua contando como um argumento embora um argumento ruim.

1.2 Sentenças

Em lógica, estamos interessados apenas em sentenças que possam aparecer como premissa ou conclusão de um argumento. Assim, diremos que uma SENTENÇA é algo que pode ser verdadeiro ou falso.

Você não deve confundir a ideia de uma sentença que pode ser verdadeira ou falsa com a diferença entre fato e opinião. Com frequência, as sentenças em lógica expressarão coisas que normalmente contaríamos como fatos como 'Kierkegaard era corcunda' ou 'Kierkegaard gostava de amêndoas'. Elas também podem expressar coisas que você talvez considere como questões de opinião como 'Amêndoas são deliciosas.'

Além disso, há coisas que contariam como 'sentenças' em um curso de linguística ou gramática e que nós não consideraremos como sentenças em lógica.

Perguntas Em uma aula de gramática, 'Você já está com sono?' contaria como uma sentença interrogativa. Embora você possa estar com sono ou acordado, a própria pergunta não é nem verdadeira nem falsa. Por essa razão, perguntas não contarão como sentenças em lógica. Suponha que você responda à pergunta: 'Eu não estou com sono.' Isso é verdadeiro ou falso, e portanto é uma sentença no sentido lógico. Em geral, *perguntas* não contam como sentenças, mas *respostas* sim.

'Sobre o que é este curso?' não é uma sentença. 'Ninguém sabe sobre o que é este curso' é uma sentença.

Imperativos Ordens são muitas vezes formuladas como imperativos, como 'Acorde!', 'Sente-se direito' e assim por diante. Em uma aula de gramática, essas contariam como sentenças imperativas. Embora possa ser bom ou não você sentar-se direito, a ordem em si não é nem verdadeira nem falsa. Note, porém, que comandos nem sempre são formulados como imperativos. 'Você vai respeitar minha autoridade' *é* verdadeira ou falsa ou você vai, ou não vai e por isso conta como uma sentença no sentido lógico.

Exclamações 'Ai!' às vezes é chamado de sentença exclamativa, mas não é nem verdadeiro nem falso. Trataremos 'Ai, machuquei meu dedo!' como significando a mesma coisa que 'Machuquei meu dedo.' O 'ai' não acrescenta nada que possa ser verdadeiro ou falso.

1.3 Duas maneiras pelas quais argumentos podem falhar

Considere o argumento de que você deveria levar um guarda-chuva (na p. 5, acima). Se a premissa (1) for falsa se estiver ensolarado então o argumento não lhe dá razão alguma para carregar um guarda-chuva. Mesmo que esteja chovendo, você pode não precisar de um guarda-chuva. Você pode estar usando uma capa de chuva, ou pode andar apenas por passagens cobertas. Nesses casos, a premissa (2) seria falsa, já que você poderia sair sem guarda-chuva e ainda assim evitar ficar encharcado.

Suponha, por um momento, que ambas as premissas sejam verdadeiras. Você não tem capa de chuva. Você precisa ir a lugares onde não há passagens cobertas. O argumento mostra então que você deve levar um guarda-chuva? Não necessariamente. Talvez você goste de caminhar na chuva e queira ficar encharcado. Nesse caso, mesmo que as premissas sejam verdadeiras, a conclusão seria falsa.

Para qualquer argumento, há duas maneiras pelas quais ele pode ser fraco. Primeiro, uma ou mais premissas podem ser falsas. Um argumento só lhe dá razão para acreditar na conclusão se você aceitar as premissas. Segundo, as premissas podem falhar em apoiar a conclusão. Mesmo que as premissas sejam verdadeiras, a forma do argumento pode ser fraca. O exemplo que acabamos de considerar é fraco nos dois sentidos.

Quando um argumento é fraco no segundo sentido, há algo errado com a *forma lógica* do argumento: premissas desse tipo não levam necessariamente a uma conclusão desse tipo. Estaremos interessados principalmente na forma lógica dos argumentos.

Considere outro exemplo:

Você está lendo este livro.
Este é um livro de lógica.
∴ Você é estudante de lógica.

Este não é um argumento terrível. A maior parte das pessoas que lê este livro é estudante de lógica. Ainda assim, é possível que alguém que não seja estudante de lógica leia este livro. Se seu colega de quarto pegar o livro e folheá-lo, ele não se torna automaticamente um estudante de lógica. Assim, as premissas desse argumento, mesmo sendo verdadeiras, não garantem a verdade da conclusão. Sua forma lógica está longe de ser perfeita.

Um argumento que não tivesse fraqueza do segundo tipo teria uma forma lógica perfeita. Se as suas premissas fossem verdadeiras, então a sua conclusão seria *necessariamente* verdadeira. Chamamos um argumento assim de 'dedutivamente válido' ou simplesmente 'válido'.

Embora possamos considerar o argumento acima como um bom argumento em certo sentido, ele não é válido; isto é, ele é 'inválido'. Uma das tarefas importantes da lógica é separar argumentos válidos de argumentos inválidos.

1.4 Validade dedutiva

Um argumento é dedutivamente VÁLIDO se, e somente se, for impossível que as premissas sejam verdadeiras e a conclusão falsa.

O ponto crucial sobre um argumento válido é que é impossível que as premissas sejam verdadeiras *ao mesmo tempo* em que a conclusão é falsa. Considere este exemplo:

Laranjas são ou frutas ou instrumentos musicais.
Laranjas não são frutas.
∴ Laranjas são instrumentos musicais.

A conclusão desse argumento é ridícula. Ainda assim, ela decorre validamente das premissas. Este é um argumento válido. *Se* ambas as premissas fossem verdadeiras, *então* a conclusão seria necessariamente verdadeira.

Isso mostra que um argumento dedutivamente válido não precisa ter premissas verdadeiras nem conclusão verdadeira. Por outro lado, ter premissas verdadeiras e conclusão verdadeira não basta para tornar um argumento válido. Considere este exemplo:

Londres fica na Inglaterra.
Pequim fica na China.
∴ Paris fica na França.

As premissas e a conclusão desse argumento são, de fato, todas verdadeiras. No entanto, este é um argumento péssimo, porque as premissas não têm nada a ver com a conclusão. Imagine o que aconteceria se Paris declarasse independência do restante da França. Então a conclusão seria falsa, embora ambas as premissas continuassem verdadeiras. Assim, é *logicamente possível* que as premissas desse argumento sejam verdadeiras e a conclusão falsa. O argumento é inválido.

A coisa importante a lembrar é que a validade não diz respeito à verdade ou falsidade efetivas das sentenças no argumento. Em vez disso, ela diz respeito à forma do argumento: a verdade das premissas é incompatível com a falsidade da conclusão.

Argumentos indutivos

Podem existir bons argumentos que, ainda assim, não são dedutivamente válidos. Considere este:

Em janeiro de 1997, choveu em San Diego.
Em janeiro de 1998, choveu em San Diego.
Em janeiro de 1999, choveu em San Diego.
∴ Chove todo mês de janeiro em San Diego.

Este é um argumento INDUTIVO, porque ele generaliza a partir de muitos casos para uma conclusão sobre todos os casos.

Certamente, o argumento poderia ser fortalecido adicionando outras premissas: em janeiro de 2000, choveu em San Diego; em janeiro de 2001, choveu em San Diego; e assim por diante. Não importa quantas premissas acrescentemos, contudo, o argumento ainda não será dedutivamente válido. É possível, embora improvável, que não chova em San Diego no próximo mês de janeiro. Além disso, sabemos que o clima pode ser caprichoso. Nenhuma quantidade de evidência deveria nos convencer de que chove lá em *todo* janeiro. Quem pode garantir que não haverá algum ano excepcional em que não chova em janeiro em San Diego? Um único contraexemplo já basta para tornar falsa a conclusão do argumento.

Argumentos indutivos, mesmo bons argumentos indutivos, não são dedutivamente válidos. Não estaremos interessados em argumentos indutivos neste livro.

1.5 Outras noções lógicas

Além da validade dedutiva, estaremos interessados em alguns outros conceitos lógicos.

Valores de verdade

Verdadeiro ou falso é o que se chama o VALOR DE VERDADE de uma sentença. Definimos sentenças como coisas que podem ser verdadeiras ou falsas; poderíamos ter dito, em vez disso, que sentenças são coisas que podem ter valores de verdade.

Verdade lógica

Ao considerar argumentos formalmente, nos preocupamos com o que seria verdadeiro *se* as premissas fossem verdadeiras. Em geral, não nos interessa o valor de verdade efetivo de sentenças particulares se elas são *de fato* verdadeiras ou falsas. Ainda assim, há sentenças que precisam ser verdadeiras, simplesmente por uma questão de lógica.

Considere estas sentenças:

1. Está chovendo.
2. Ou está chovendo, ou não está.
3. Está chovendo e não está chovendo ao mesmo tempo.

Para saber se a sentença 1 é verdadeira, você precisaria olhar pela janela ou consultar a previsão do tempo. Do ponto de vista lógico, ela pode ser verdadeira ou falsa. Sentenças como essa são chamadas de sentenças *contingentes*.

A sentença 2 é diferente. Você não precisa olhar para fora para saber que ela é verdadeira. Independentemente de como estiver o tempo, ou está chovendo ou não está. Essa sentença é *logicamente verdadeira*; ela é verdadeira apenas em virtude da lógica, não importando como o mundo de fato seja. Uma sentença logicamente verdadeira é chamada de TAUTOLOGIA.

Você também não precisa verificar o tempo para decidir a respeito da sentença 3. Ela precisa ser falsa, simplesmente por uma questão de lógica. Pode estar chovendo aqui e não chovendo em outra cidade; pode estar chovendo agora e parar de chover enquanto você lê isto; mas é impossível que esteja ao mesmo tempo chovendo e não chovendo aqui, neste exato momento. A terceira sentença é *logicamente falsa*; ela é falsa independentemente de como o mundo seja. Uma sentença logicamente falsa é chamada de CONTRADIÇÃO.

Para sermos precisos, podemos definir uma SENTENÇA CONTINGENTE como uma sentença que não é nem uma tautologia nem uma contradição.

Uma sentença pode ser *sempre* verdadeira e ainda assim ser contingente. Por exemplo, se nunca houve um momento em que o universo tivesse menos do que sete coisas, então a sentença 'Existem pelo menos sete coisas' será sempre verdadeira. Mesmo assim, a sentença é contingente; a sua verdade não é uma questão de lógica. Não há contradição em considerar um mundo possível em que existam menos que sete coisas. A questão importante é se a sentença *precisa* ser verdadeira, apenas em virtude da lógica.

Equivalência lógica

Também podemos perguntar sobre as relações lógicas *entre* duas sentenças. Por exemplo:

> João foi ao mercado depois de lavar a louça.
> João lavou a louça antes de ir ao mercado.

Essas duas sentenças são ambas contingentes, já que João poderia não ter ido ao mercado nem lavado a louça. Ainda assim, elas precisam ter o mesmo valor de verdade. Se uma delas for verdadeira, então a outra também é; se uma delas for falsa, então a outra também é. Quando duas sentenças necessariamente têm o mesmo valor de verdade, dizemos que elas são LOGICAMENTE EQUIVALENTES.

Consistência

Considere estas duas sentenças:

> **B1** Meu único irmão é mais alto do que eu.
> **B2** Meu único irmão é mais baixo do que eu.

A lógica, por si só, não pode nos dizer qual dessas sentenças é verdadeira, se é que alguma delas é. Ainda assim, podemos dizer que *se* a primeira sentença (B1) for verdadeira, *então* a segunda (B2) deve ser falsa. E se B2 for verdadeira, então B1 deve ser falsa. Não pode acontecer de ambas as sentenças serem verdadeiras ao mesmo tempo.

Se um conjunto de sentenças não puder ser verdadeiro em sua totalidade, como B1B2, dizemos que ele é INCONSISTENTE. Caso contrário, dizemos que é CONSISTENTE.

Podemos perguntar sobre a consistência de qualquer quantidade de sentenças. Por exemplo, considere a seguinte lista:

G1 Há pelo menos quatro girafas no parque de animais selvagens.
G2 Há exatamente sete gorilas no parque de animais selvagens.
G3 Não há mais do que dois marcianos no parque de animais selvagens.
G4 Toda girafa no parque de animais selvagens é marciana.

G1 e G4, juntas, implicam que há pelo menos quatro girafas marcianas no parque. Isso entra em conflito com G3, que implica que não há mais do que duas girafas marcianas lá. Então o conjunto de sentenças G1G4 é inconsistente. Note que a inconsistência não tem nada a ver com G2. G2 apenas acaba fazendo parte de um conjunto inconsistente.

Às vezes, as pessoas dizem que um conjunto inconsistente de sentenças 'contém uma contradição'. Com isso, querem dizer que seria logicamente impossível que todas as sentenças fossem verdadeiras ao mesmo tempo. Um conjunto pode ser inconsistente mesmo que cada sentença nele seja contingente ou tautológica. Quando uma única sentença é uma contradição, então essa sentença sozinha não pode ser verdadeira.

1.6 Linguagens formais

Aqui está um argumento famoso e válido:

Sócrates é um homem.
Todos os homens são mortais.
$\therefore$ Sócrates é mortal.

Este é um argumento irrefutável. A única maneira de contestar a conclusão é negando uma das premissas a forma lógica é impecável. E quanto ao argumento seguinte?

Sócrates é um homem.
Todos os homens são cenouras.
$\therefore$ Sócrates é uma cenoura.

Esse argumento talvez seja menos interessante que o primeiro, porque a segunda premissa é obviamente falsa. Não há nenhum sentido claro em que todos os homens sejam cenouras. Ainda assim, o argumento é válido. Para ver isso, note que ambos os argumentos têm a seguinte forma:

S é M.
Todos os Ms são Cs.
$\therefore$ S é C.

Nos dois argumentos, S representa Sócrates e M representa homem. No primeiro argumento, C representa mortal; no segundo, C representa cenoura. Ambos os argumentos têm essa forma, e todo argumento dessa forma é válido. Logo, ambos os argumentos são válidos.

O que fizemos aqui foi substituir palavras como 'homem' ou 'cenoura' por símbolos como 'M' ou 'C' para tornar explícita a forma lógica. Essa é a ideia central por trás da lógica formal. Queremos remover aspectos irrelevantes ou distrações do argumento para tornar a forma lógica mais transparente.

Partindo de um argumento em uma *linguagem natural* como o inglês, traduzimos o argumento para uma *linguagem formal*. Partes das sentenças em inglês são substituídas por letras e símbolos. O objetivo é revelar a estrutura formal do argumento, como fizemos com esses dois exemplos.

Existem linguagens formais que funcionam de modo semelhante à simbolização que demos para esses dois argumentos. Uma lógica desse tipo foi desenvolvida por Aristóteles, um filósofo que viveu na Grécia no século IV a.C. Aristóteles foi aluno de Platão e tutor de Alexandre, o Grande. A lógica aristotélica, com algumas revisões, foi a lógica dominante no mundo ocidental por mais de dois milênios.

Na lógica aristotélica, categorias são representadas por letras maiúsculas. Cada sentença de um argumento é então representada como tendo uma das quatro formas, que os lógicos medievais nomearam assim: (A) Todos os As são Bs. (E) Nenhum A é B. (I) Algum A é B. (O) Algum A não é B.

Isso permite descrever *silogismos* válidos, isto é, argumentos de três linhas como os dois que consideramos acima. Lógicos medievais deram nomes mnemônicos a todas as formas de argumento válidas. A forma dos nossos dois argumentos, por exemplo, era chamada de *Barbara*. As vogais no nome, todas As, indicam que as duas premissas e a conclusão são sentenças da forma (A).

Existem muitas limitações na lógica aristotélica. Uma delas é que ela não distingue claramente entre tipos (espécies, classes) e indivíduos. Assim, a primeira premissa poderia ser escrita igualmente como 'Todos os Ss são Ms': todos os Sócrates são homens. Apesar de sua importância histórica, a lógica aristotélica foi superada. O restante deste livro desenvolverá duas linguagens formais.

A primeira é SL, que significa *lógica sentencial* (*sentential logic*). Em SL, as menores unidades são as próprias sentenças. Sentenças simples são representadas por letras e conectadas por conectivos lógicos como 'e' e 'não' para formar sentenças mais complexas.

A segunda é QL, que significa *lógica quantificada* (*quantified logic*). Em QL, as unidades básicas são objetos, propriedades de objetos e relações entre objetos.

Quando traduzimos um argumento para uma linguagem formal, esperamos tornar sua estrutura lógica mais clara. Queremos incluir o suficiente da estrutura do argumento em língua inglesa para podermos julgar se o argumento é válido ou inválido. Se incluíssemos todo o conteúdo da linguagem inglesa, com toda a sua sutileza e nuance, então não haveria vantagem em traduzir para uma linguagem formal. Seria melhor simplesmente pensar sobre o argumento diretamente em inglês.

Ao mesmo tempo, gostaríamos de ter uma linguagem formal que nos permita representar muitos tipos diferentes de argumentos em inglês. Essa é uma razão para preferir QL à lógica aristotélica; QL pode representar todos os argumentos válidos da lógica aristotélica e ainda mais.

Assim, ao decidir sobre uma linguagem formal, há inevitavelmente uma tensão entre querer captar o máximo possível de estrutura e querer uma linguagem formal simples linguagens formais mais simples deixam de fora mais detalhes. Isso significa que não existe uma linguagem formal perfeita. Algumas farão um trabalho melhor do que outras ao traduzir certos tipos de argumentos em linguagem natural.

Neste livro, assumimos que *verdadeiro* e *falso* são os únicos valores de verdade possíveis. Linguagens lógicas que fazem essa suposição são chamadas de *bivalentes*, o que significa *com dois valores*. A lógica aristotélica, SL e QL são todas bivalentes, mas há limites para o poder de lógicas bivalentes. Por exemplo, alguns filósofos afirmaram que o futuro ainda não está determinado. Se eles estiverem certos, então sentenças sobre *o que virá a ser o caso* ainda não são verdadeiras nem falsas. Algumas linguagens formais levam isso em conta permitindo sentenças que não são nem verdadeiras nem falsas, mas algo intermediário. Outras linguagens formais, as chamadas lógicas paraconsistentes, permitem sentenças que são ao mesmo tempo verdadeiras *e* falsas.

As linguagens apresentadas neste livro não são as únicas linguagens formais possíveis. No entanto, a maior parte das lógicas não padrão estende a estrutura formal básica das lógicas bivalentes discutidas aqui. Por isso, este é um bom lugar para começar.

Resumo das noções lógicas

- ▷ Um argumento é (dedutivamente) VÁLIDO se for impossível que as premissas sejam verdadeiras e a conclusão falsa; é INVÁLIDO caso contrário.
- ▷ Uma TAUTOLOGIA é uma sentença que, por força da lógica, deve ser verdadeira.
- ▷ Uma CONTRADIÇÃO é uma sentença que, por força da lógica, deve ser falsa.
- ▷ Uma SENTENÇA CONTINGENTE não é nem uma tautologia nem uma contradição.
- ▷ Duas sentenças são LOGICAMENTE EQUIVALENTES se necessariamente tiverem o mesmo valor de verdade.
- ▷ Um conjunto de sentenças é CONSISTENTE se for logicamente possível que todos os membros do conjunto sejam verdadeiros ao mesmo tempo; é INCONSISTENTE caso contrário.

Practice Exercises

Ao final de cada capítulo, você encontrará uma série de exercícios que revisam e exploram o conteúdo tratado no capítulo. Não há substituto para realmente resolver problemas, porque lógica diz mais respeito a um modo de pensar do que a memorizar fatos. As respostas de alguns dos exercícios são fornecidas ao final do livro, no apêndice B; os exercícios que têm solução no apêndice são marcados com um $\star$.

Part A Quais das sentenças a seguir são ‘sentenças’ no sentido lógico?

1. A Inglaterra é menor do que a China.
2. A Groenlândia fica ao sul de Jerusalém.
3. Nova Jersey fica a leste de Wisconsin?
4. O número atômico do hélio é 2.
5. O número atômico do hélio é π.
6. Eu odeio macarrão passado do ponto.
7. Eca! Macarrão passado do ponto!
8. Macarrão passado do ponto é nojento.
9. Vá com calma.
10. Esta é a última questão.

Part B Para cada uma das sentenças a seguir: ela é uma tautologia, uma contradição ou uma sentença contingente?

1. César atravessou o Rubicão.
2. Alguém já atravessou o Rubicão.
3. Ninguém jamais atravessou o Rubicão.
4. Se César atravessou o Rubicão, então alguém o atravessou.
5. Embora César tenha atravessado o Rubicão, ninguém jamais atravessou o Rubicão.
6. Se alguém já atravessou o Rubicão, então foi César.

⋆ **Part C** Retome as sentenças G1G4 na p. 12 e considere cada um dos seguintes conjuntos de sentenças. Quais são consistentes? Quais são inconsistentes?

1. G2, G3 e G4
2. G1, G3 e G4
3. G1, G2 e G4
4. G1, G2 e G3

⋆ **Part D** Quais das situações a seguir são possíveis? Se for possível, dê um exemplo. Se não for possível, explique por quê.

1. Um argumento válido que tenha uma premissa falsa e uma premissa verdadeira
2. Um argumento válido que tenha uma conclusão falsa
3. Um argumento válido cuja conclusão seja uma contradição
4. Um argumento inválido cuja conclusão seja uma tautologia
5. Uma tautologia que seja contingente
6. Duas sentenças logicamente equivalentes, ambas tautologias
7. Duas sentenças logicamente equivalentes, uma das quais é uma tautologia e a outra é contingente
8. Duas sentenças logicamente equivalentes que, juntas, formem um conjunto inconsistente
9. Um conjunto consistente de sentenças que contenha uma contradição
10. Um conjunto inconsistente de sentenças que contenha uma tautologia

Chapter 2

Lógica sentencial

Este capítulo apresenta uma linguagem lógica chamada SL. Ela é uma versão de *lógica sentencial*, porque as unidades básicas da linguagem vão representar sentenças inteiras.

2.1 Letras de sentença

Em SL, letras maiúsculas são usadas para representar sentenças básicas. Considerada apenas como um símbolo de SL, a letra A pode significar qualquer sentença. Portanto, ao traduzir do inglês para SL, é importante fornecer uma *chave de simbolização*. A chave associa a cada letra de sentença usada uma sentença em linguagem natural (aqui, originalmente, em inglês).

Por exemplo, considere este argumento:

Há uma maçã sobre a mesa.
Se há uma maçã sobre a mesa, então Jenny chegou à aula.
∴ Jenny chegou à aula.

Este é obviamente um argumento válido em inglês. Ao simbolizá-lo, queremos preservar a estrutura do argumento que o torna válido. O que acontece se simplesmente substituirmos cada sentença por uma letra? Nossa chave de simbolização ficaria assim:

A: Há uma maçã sobre a mesa.
B: Se há uma maçã sobre a mesa, então Jenny chegou à aula.
C: Jenny chegou à aula.

Então simbolizaríamos o argumento da seguinte forma:

A
B
$\therefore C$

Não há conexão necessária entre alguma sentença A, que poderia ser qualquer sentença, e outras sentenças B e C, que também poderiam ser quaisquer sentenças. A estrutura do argumento foi completamente perdida nessa tradução.

O ponto importante sobre o argumento é que a segunda premissa não é apenas *uma* sentença qualquer, logicamente desconectada das outras. A segunda premissa contém a primeira premissa e a conclusão *como partes*. Nossa chave de simbolização para o argumento só precisa incluir significados para A e C, e podemos construir a segunda premissa a partir dessas peças. Assim, simbolizamos o argumento desta forma:

A
Se A, então C.
$\therefore C$

Isso preserva a estrutura do argumento que o torna válido, mas ainda faz uso da expressão em inglês 'If… then…'. Embora queiramos, ao final, substituir todas as expressões do inglês por notação lógica, este já é um bom começo.

As sentenças que podem ser simbolizadas por letras de sentença são chamadas de *sentenças atômicas*, porque elas são os blocos básicos a partir dos quais sentenças mais complexas podem ser construídas. Qualquer estrutura lógica interna de uma sentença é perdida quando ela é traduzida como sentença atômica. Do ponto de vista de SL, a sentença vira apenas uma letra. Ela pode ser usada para construir sentenças mais complexas, mas não pode ser desmontada.

Há apenas vinte e seis letras no alfabeto, mas não há limite lógico para o número de sentenças atômicas. Podemos usar a mesma letra para simbolizar sentenças atômicas diferentes acrescentando um subscrito, um pequeno número escrito após a letra. Assim, poderíamos ter uma chave de simbolização como esta:

$\mathbf{A}_1$: A maçã está embaixo do armário.
$\mathbf{A}_2$: Argumentos em SL sempre contêm sentenças atômicas.
$\mathbf{A}_3$: Adam Ant está pegando um avião de Anchorage para Albany.
$\vdots$
$\mathbf{A}_{294}$: Aliterações aborrecem astronautas afáveis.

Lembre-se de que cada uma dessas é uma letra de sentença diferente. Quando há subscritos na chave de simbolização, é importante acompanhá-los com cuidado.

2.2 Conectivos

Conectivos lógicos são usados para construir sentenças complexas a partir de componentes atômicos. Há cinco conectivos lógicos em SL. A tabela abaixo os resume; em seguida, cada um é explicado.

símbolo	como é chamado	o que significa
$\neg$	negação	'Não é o caso que...'
&	conjunção	'Tanto... quanto ...'
$\vee$	disjunção	'Ou... ou ...'
$\rightarrow$	condicional	'Se ... então ...'
$\leftrightarrow$	bicondicional	'... se e somente se ...'

Negação

Considere como poderíamos simbolizar estas sentenças:

1. Mary está em Barcelona.
2. Mary não está em Barcelona.
3. Mary está em algum lugar diferente de Barcelona.

Para simbolizar a sentença 1, precisamos de uma letra de sentença. Podemos fornecer a seguinte chave de simbolização:

B: Mary está em Barcelona.

Observe que aqui estamos dando a B uma interpretação diferente daquela usada na seção anterior. A chave de simbolização só especifica o que B significa *em um contexto específico*. É fundamental que continuemos a usar esse significado de B enquanto estivermos falando sobre Mary e Barcelona. Mais tarde, ao simbolizar sentenças diferentes, podemos escrever uma nova chave de simbolização e usar B para significar outra coisa.

Agora, a sentença 1 é simplesmente B.

Como a sentença 2 é obviamente relacionada à sentença 1, não queremos introduzir outra letra de sentença. Em parte em português, a sentença significa não B. Para simbolizá-la, precisamos de um símbolo para a negação lógica. Usaremos '$\neg$'. Assim, podemos traduzir não B por $\neg B$.

A sentença 3 fala sobre se Mary está ou não em Barcelona, embora não contenha a palavra não. Ainda assim, ela é claramente logicamente equivalente à sentença 2. Ambas significam: Não é o caso que Mary está em Barcelona. Portanto, podemos traduzir tanto a sentença 2 quanto a 3 como $\neg B$.

> Uma sentença pode ser simbolizada como $\neg\mathcal{A}$ se puder ser parafraseada em português como 'Não é o caso que $\mathcal{A}$.'

Considere agora estes exemplos:

4. A peça pode ser substituída se quebrar.
5. A peça é insubstituível.
6. A peça não é insubstituível.

Se deixarmos R significar 'A peça é substituível', então a sentença 4 pode ser traduzida como R.

E quanto à sentença 5? Dizer que a peça é insubstituível significa que não é o caso que a peça é substituível. Assim, embora a sentença 5 não seja negativa em português, nós a simbolizamos usando negação: $\neg R$.

A sentença 6 pode ser parafraseada como 'Não é o caso que a peça é insubstituível.' Usando negação duas vezes, traduzimos isso como $\neg\neg R$. As duas negações em sequência funcionam cada uma como negação, de modo que a sentença significa não é o caso que... não é o caso que... R. Pensando em português, ela é logicamente equivalente à sentença 4. Assim, quando definirmos equivalência lógica em SL, faremos com que R e $\neg\neg R$ sejam logicamente equivalentes.

Mais exemplos:

7. Elliott está feliz.
8. Elliott está infeliz.

Se deixarmos H significar 'Elliott está feliz', então podemos simbolizar a sentença 7 como H.

No entanto, seria um erro simbolizar a sentença 8 como $\neg H$. Se Elliott está infeliz, então ele não está feliz mas a sentença 8 não significa o mesmo que não é o caso que Elliott está feliz. Pode ser que ele não esteja feliz, mas também não esteja infeliz; talvez esteja em algum ponto intermediário. Para permitir a possibilidade de indiferença, precisamos de uma nova letra de sentença para simbolizar 8.

Para qualquer sentença $\mathcal{A}$: se $\mathcal{A}$ é verdadeira, então $\neg\mathcal{A}$ é falsa. Se $\neg\mathcal{A}$ é verdadeira, então $\mathcal{A}$ é falsa. Usando 'T' para verdadeiro e 'F' para falso, podemos resumir isso numa *tabela de verdade característica* para a negação:

$\mathcal{A}$	$\neg\mathcal{A}$
T	F
F	T

Falaremos mais detalhadamente de tabelas de verdade no próximo capítulo.

Conjunção

Considere estas sentenças:

9. Adam é atlético.
10. Barbara é atlética.
11. Adam é atlético, e Barbara também é atlética.

Precisaremos de letras de sentença distintas para 9 e 10, então definimos esta chave de simbolização:

A: Adam é atlético.
B: Barbara é atlética.

A sentença 9 pode ser simbolizada como A.

A sentença 10 pode ser simbolizada como B.

A sentença 11 pode ser parafraseada como 'A e B'. Para simbolizá-la completamente, precisamos de outro símbolo. Usaremos '&'. Traduzimos então 'A e B' como $A \,\&\, B$. O conectivo lógico '&' é chamado de CONJUNÇÃO, e A e B são chamados de CONJUNTOS (ou conjunções parciais).

Observe que não tentamos simbolizar a palavra 'também' em 11. Palavras como 'tanto', 'ambos' e 'também' servem apenas para chamar a atenção para o fato de que duas coisas estão sendo conjuntadas. Elas não têm função lógica adicional, então não precisamos representá-las em SL.

Mais alguns exemplos:

12. Barbara é atlética e energética.
13. Barbara e Adam são ambos atléticos.
14. Embora Barbara seja energética, ela não é atlética.
15. Barbara é atlética, mas Adam é mais atlético do que ela.

A sentença 12 é claramente uma conjunção. A sentença diz duas coisas sobre Barbara; em português é permitido mencionar Barbara apenas uma vez. Poderia ser tentador traduzir assim: já que B significa 'Barbara é atlética', alguém poderia parafrasear como 'B e energética'. Isso seria um erro. Uma vez que traduzimos parte da sentença como B, qualquer estrutura interna é perdida. B é uma sentença atômica; não é mais do que verdadeira ou falsa. Por outro lado, 'energética' não é uma sentença; sozinha não é nem verdadeira nem falsa. Devemos, ao contrário, parafrasear a sentença como 'B e Barbara é energética.' Agora precisamos acrescentar uma letra de sentença à chave de simbolização. Seja E 'Barbara é energética'. Agora a sentença pode ser traduzida como $B \,\&\, E$.

Uma sentença pode ser simbolizada como $\mathcal{A} \,\&\, \mathcal{B}$ se puder ser parafraseada em português como 'Tanto $\mathcal{A}$ quanto $\mathcal{B}$.' Cada um dos conjuntos deve ser uma sentença.

A sentença 13 afirma uma coisa sobre dois sujeitos distintos. Ela diz, tanto de Barbara quanto de Adam, que são atléticos, e em português usamos a palavra 'atléticos' apenas uma vez. Ao traduzir para SL, é importante perceber que a sentença pode ser parafraseada como 'Barbara é atlética, e Adam é atlético.' Isso se traduz como $B \,\&\, A$.

A sentença 14 é um pouco mais complicada. A palavra 'embora' estabelece um contraste entre a primeira parte da sentença e a segunda. Ainda assim, a sentença diz tanto que Barbara é energética quanto que ela não é atlética. Para fazer com que cada um dos conjuntos seja uma sentença atômica, precisamos substituir 'ela' por 'Barbara'.

Podemos então parafrasear 14 como '*Tanto* Barbara é energética *como* Barbara não é atlética.' A segunda conjunção contém uma negação, então podemos parafrasear ainda mais: '*Tanto* Barbara é energética *como não é o caso que* Barbara é atlética.' Isso se traduz como $E \,\&\, \neg B$.

A sentença 15 tem uma estrutura contrastiva semelhante. Isso é irrelevante para a tradução em SL, então podemos parafraseá-la como '*Tanto* Barbara é atlética, *como* Adam é mais atlético do que Barbara.' (Observe que novamente substituímos o pronome 'ela' pelo nome.) Como traduzir o segundo conjunto? Já temos a letra A que fala de Adam ser atlético e B que fala de Barbara ser atlética, mas nenhuma delas fala de um ser mais atlético do que o outro. Precisamos de uma nova letra de sentença. Seja R 'Adam é mais atlético do que Barbara.' Agora a sentença se traduz como $B \,\&\, R$.

Sentenças que podem ser parafraseadas como '$\mathcal{A}$, mas $\mathcal{B}$' ou 'Embora $\mathcal{A}$, $\mathcal{B}$' são melhor simbolizadas usando conjunção: $\mathcal{A} \,\&\, \mathcal{B}$

É importante lembrar que as letras de sentença A, B e R são sentenças atômicas. Consideradas como símbolos de SL, elas não têm significado além de serem verdadeiras ou falsas. Nós as usamos para simbolizar sentenças em português que, todas, falam de pessoas atléticas; mas essa semelhança é completamente perdida quando traduzimos para SL. Nenhuma linguagem formal consegue capturar toda a estrutura da linguagem natural, mas, enquanto essa estrutura extra não for importante para o argumento, nada é perdido ao deixá-la de lado.

Para quaisquer sentenças $\mathcal{A}$ e $\mathcal{B}$, $\mathcal{A} \,\&\, \mathcal{B}$ é verdadeira se, e somente se, tanto $\mathcal{A}$ quanto $\mathcal{B}$ forem verdadeiras. Podemos resumir isso na tabela de verdade característica para a conjunção:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \,\&\, \mathcal{B}$
T	T	T
T	F	F
F	T	F
F	F	F

A conjunção é *simétrica*, porque podemos trocar os conjuntos sem mudar o valor de verdade da sentença. Quaisquer que sejam $\mathcal{A}$ e $\mathcal{B}$, $\mathcal{A} \,\&\, \mathcal{B}$ é logicamente equivalente a $\mathcal{B} \,\&\, \mathcal{A}$.

Disjunção

Considere estas sentenças:

16. Ou Denison vai jogar golfe comigo, ou ele vai assistir a filmes.
17. Ou Denison ou Ellery vai jogar golfe comigo.

Para essas sentenças podemos usar a seguinte chave de simbolização:

D: Denison vai jogar golfe comigo.
E: Ellery vai jogar golfe comigo.
M: Denison vai assistir a filmes.

A sentença 16 é 'Ou D ou M.' Para simbolizá-la completamente, introduzimos um novo símbolo. A sentença torna-se $D \vee M$. O conectivo '$\vee$' é chamado de DISJUNÇÃO, e D e M são chamados de DISJUNTOS.

A sentença 17 é apenas um pouco mais complicada. Há dois sujeitos, mas a sentença em português só apresenta o verbo uma vez. Ao traduzir, podemos parafraseá-la como 'Ou Denison vai jogar golfe comigo, ou Ellery vai jogar golfe comigo.' Agora, ela claramente se traduz como $D \vee E$.

> Uma sentença pode ser simbolizada como $\mathcal{A} \vee \mathcal{B}$ se puder ser parafraseada em português como 'Ou $\mathcal{A}$, ou $\mathcal{B}$.' Cada disjunto deve ser uma sentença.

Às vezes, em português, a palavra 'ou' exclui a possibilidade de os dois disjuntos serem verdadeiros. Isso é chamado de OU EXCLUSIVO. Um *ou exclusivo* é claramente pretendido quando, em um cardápio, se diz: 'Os pratos principais vêm com sopa ou salada.' Você pode escolher sopa; pode escolher salada; mas, se quiser *tanto* sopa *quanto* salada, precisará pagar a mais.

Em outras situações, a palavra 'ou' permite a possibilidade de ambos os disjuntos serem verdadeiros. Provavelmente é o caso em 17, acima. Eu posso jogar com Denison, com Ellery, ou com ambos. A sentença 17 apenas diz que vou jogar com *pelo menos* um deles. Isso é chamado de OU INCLUSIVO.

O símbolo '$\vee$' representa um *ou inclusivo*. Assim, $D \vee E$ é verdadeiro se D é verdadeiro, se E é verdadeiro, ou se tanto D quanto E são verdadeiros. Ele é falso apenas se tanto D quanto E forem falsos. Podemos resumir isso na tabela de verdade característica da disjunção:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$
T	T	T
T	F	T
F	T	T
F	F	F

Assim como a conjunção, a disjunção é simétrica. $\mathcal{A}\vee\mathcal{B}$ é logicamente equivalente a $\mathcal{B}\vee\mathcal{A}$.

Estas sentenças são um pouco mais complicadas:

18. Ou você não vai tomar sopa, ou você não vai tomar salada.
19. Você não vai tomar nem sopa nem salada.
20. Você ganha sopa ou salada, mas não as duas.

Seja S_1 'você toma sopa' e S_2 'você toma salada'.

A sentença 18 pode ser parafraseada assim: 'Ou *não é o caso que* você toma sopa, ou *não é o caso que* você toma salada.' Traduzir isso exige disjunção e negação. Fica $\neg S_1 \vee \neg S_2$.

A sentença 19 também exige negação. Ela pode ser parafraseada como '*Não é o caso que* (você toma sopa ou você toma salada).' Precisamos de alguma forma indicar que a negação não está apenas negando o disjunto da direita ou da esquerda, mas sim a disjunção inteira. Para isso, colocamos parênteses em torno da disjunção: 'Não é o caso que $(S_1 \vee S_2)$.' Isso se torna simplesmente $\neg(S_1 \vee S_2)$.

Observe que os parênteses fazem um trabalho importante aqui. A sentença $\neg S_1 \vee S_2$ significaria 'Ou você não toma sopa, ou você toma salada.'

A sentença 20 é um *ou exclusivo*. Podemos decompor a sentença em duas partes. A primeira parte diz que você ganha uma coisa ou outra. Traduzimos isso como $(S_1 \vee S_2)$. A segunda parte diz que você não ganha ambas. Podemos parafraseá-la como 'Não é o caso que você toma sopa e salada.' Usando negação e conjunção, traduzimos isso como $\neg(S_1 \,\&\, S_2)$. Agora só falta juntar as duas partes. Como vimos antes, 'mas' geralmente pode ser traduzido como conjunção. Assim, 20 pode ser traduzida como $(S_1 \vee S_2) \,\&\, \neg(S_1 \,\&\, S_2)$.

Embora '$\vee$' seja um *ou inclusivo*, podemos simbolizar um *ou exclusivo* em SL. Só precisamos de mais de um conectivo para fazê-lo.

Condicional

Para as sentenças a seguir, deixe R significar 'Você vai cortar o fio vermelho' e B significar 'A bomba vai explodir.'

21. Se você cortar o fio vermelho, então a bomba vai explodir.
22. A bomba vai explodir somente se você cortar o fio vermelho.

A sentença 21 pode ser parcialmente traduzida como 'Se R, então B.' Usaremos o símbolo '$\rightarrow$' para representar o condicional lógico. A sentença se torna $R \rightarrow B$. O conectivo é chamado de CONDICIONAL. A sentença à esquerda do condicional (R neste exemplo) é chamada de ANTECEDENTE. A sentença à direita (B) é chamada de CONSEQUENTE.

A sentença 22 também é um condicional. Como a palavra 'se' aparece na segunda metade da sentença, pode ser tentador simbolizá-la da mesma forma que a sentença 21. Isso seria um erro.

O condicional $R \rightarrow B$ diz que, *se* R for verdadeiro, *então* B também será verdadeiro. Ele não diz que cortar o fio vermelho é a *única* maneira pela qual a bomba poderia explodir; outra pessoa pode cortar o fio, ou a bomba pode estar em um temporizador. A sentença $R \rightarrow B$ não diz nada sobre o que esperar se R for falso. A sentença 22 é diferente. Ela diz que as únicas condições sob as quais a bomba explodirá envolvem você ter cortado o fio vermelho; isto é, se a bomba explodir, então você deve ter cortado o fio. Assim, 22 deve ser simbolizada como $B \rightarrow R$.

É importante lembrar que o conectivo '$\rightarrow$' diz apenas que, se o antecedente é verdadeiro, então o consequente é verdadeiro. Ele não diz nada sobre a conexão *causal* entre os dois eventos. Traduzir 22 como $B \rightarrow R$ não significa que a explosão da bomba causaria o fato de você cortar o fio. Tanto 21 quanto 22 sugerem que, se você cortar o fio vermelho, esse corte seria a causa da explosão. Elas diferem na conexão *lógica*. Se 22 fosse verdadeira, então uma explosão nos diria a nós que estamos longe da bomba que você cortou o fio vermelho. Sem explosão, 22 não nos diz nada.

> A sentença parafraseada como '$\mathcal{A}$ somente se $\mathcal{B}$' é logicamente equivalente a 'Se $\mathcal{A}$, então $\mathcal{B}$.'

'Se $\mathcal{A}$ então $\mathcal{B}$' significa que, se $\mathcal{A}$ é verdadeira, então $\mathcal{B}$ também é. Sabemos, portanto, que se o antecedente $\mathcal{A}$ for verdadeiro e o consequente $\mathcal{B}$ for falso, o condicional 'Se $\mathcal{A}$ então $\mathcal{B}$' é falso. Qual é o valor de verdade de 'Se $\mathcal{A}$ então $\mathcal{B}$' nas outras situações? Suponha, por exemplo, que o antecedente $\mathcal{A}$ seja falso. A sentença 'Se $\mathcal{A}$ então $\mathcal{B}$' não nos dirá nada sobre o valor de verdade efetivo do consequente $\mathcal{B}$, e não é óbvio qual deveria ser o valor de verdade do condicional.

Em português (e em inglês), a verdade de condicionais muitas vezes depende do que *aconteceria* se o antecedente *fosse verdadeiro* mesmo que, de fato, o antecedente seja falso. Isso cria um problema para traduzir condicionais para SL. Consideradas como sentenças de SL, R e B nos exemplos acima não têm, por si mesmas, nenhuma relação interna. Para considerar como o mundo seria se R fosse verdadeira, precisaríamos analisar o conteúdo de R. Mas, como R é um símbolo atômico de SL, não há estrutura interna a ser analisada. Ao substituir uma sentença por uma letra de sentença, passamos a considerá-la apenas como uma sentença atômica que pode ser verdadeira ou falsa.

Para traduzir condicionais em SL, não tentaremos capturar todas as sutilezas da expressão natural 'Se... então...'. Em vez disso, o símbolo '$\rightarrow$' será um *condicional material.* Isso significa que, quando $\mathcal{A}$ é falsa, o condicional $\mathcal{A}\rightarrow\mathcal{B}$ é automaticamente verdadeiro, independentemente do valor de verdade de $\mathcal{B}$. Se tanto $\mathcal{A}$ quanto $\mathcal{B}$ forem verdadeiras, então o condicional $\mathcal{A}\rightarrow\mathcal{B}$ também é verdadeiro.

Em resumo, $\mathcal{A}\rightarrow\mathcal{B}$ é falso se, e somente se, $\mathcal{A}$ é verdadeira e $\mathcal{B}$ é falsa. Podemos resumir isso com a tabela de verdade característica do condicional.

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$
T	T	T
T	F	F
F	T	T
F	F	T

O condicional é *assimétrico*. Não podemos trocar antecedente e consequente sem mudar o significado da sentença, porque $\mathcal{A}\rightarrow\mathcal{B}$ e $\mathcal{B}\rightarrow\mathcal{A}$ não são logicamente equivalentes.

Nem todas as sentenças da forma 'Se... então...' são condicionais reais. Considere esta sentença:

23. Se alguém quiser me ver, eu estarei na varanda.

Se eu digo isso, quero dizer que vou estar na varanda, independentemente de alguém querer me ver ou não mas, se alguém quiser me ver, deve me procurar lá. Se deixarmos P significar 'Eu estarei na varanda', então 23 pode ser traduzida simplesmente como P.

Bicondicional

Considere estas sentenças:

24. A figura no quadro é um triângulo somente se tiver exatamente três lados.
25. A figura no quadro é um triângulo, se tiver exatamente três lados.
26. A figura no quadro é um triângulo se e somente se tiver exatamente três lados.

Seja T 'A figura é um triângulo' e S 'A figura tem três lados.'

A sentença 24, pelos motivos discutidos acima, pode ser traduzida como $T \rightarrow S$.

A sentença 25 é importante e diferentemente construída. Ela pode ser parafraseada como 'Se a figura tem três lados, então é um triângulo.' Assim, pode ser traduzida como $S \rightarrow T$.

A sentença 26 diz que T é verdadeira *se e somente se* S é verdadeira; podemos inferir S a partir de T, e podemos inferir T a partir de S. Isso é chamado de BICONDICIONAL, porque implica os dois condicionais $S \rightarrow T$ e $T \rightarrow S$. Usaremos '$\leftrightarrow$' para representar o bicondicional; assim, 26 pode ser traduzida como $S \leftrightarrow T$.

Poderíamos viver sem um novo símbolo para o bicondicional. Como 26 significa '$T \rightarrow S$ e $S \rightarrow T$', poderíamos traduzi-la como $(T \rightarrow S) \& (S \rightarrow T)$. Precisaríamos de parênteses para indicar que $(T \rightarrow S)$ e $(S \rightarrow T)$ são conjunções separadas; a expressão $T \rightarrow S \& S \rightarrow T$ seria ambígua.

Como sempre poderíamos escrever $(\mathcal{A} \rightarrow \mathcal{B}) \& (\mathcal{B} \rightarrow \mathcal{A})$ no lugar de $\mathcal{A} \leftrightarrow \mathcal{B}$, não *precisaríamos*, em sentido estrito, introduzir um novo símbolo para o bicondicional. Ainda assim, linguagens lógicas geralmente têm esse símbolo. SL terá um, o que torna mais simples traduzir expressões como 'se e somente se'.

$\mathcal{A} \leftrightarrow \mathcal{B}$ é verdadeira se, e somente se, $\mathcal{A}$ e $\mathcal{B}$ tiverem o mesmo valor de verdade. Esta é a tabela de verdade característica do bicondicional:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \leftrightarrow \mathcal{B}$
T	T	T
T	F	F
F	T	F
F	F	T

2.3 Outras simbolizações

Agora já introduzimos todos os conectivos de SL. Podemos usá-los em conjunto para traduzir muitos tipos de sentenças. Considere estes exemplos de sentenças com o conectivo em português ‘a menos que’:

27. A menos que você vista um casaco, vai pegar um resfriado.
28. Você vai pegar um resfriado, a menos que vista um casaco.

Seja J ‘Você vai vestir um casaco’ e D ‘Você vai pegar um resfriado.’

Podemos parafrasear 27 como ‘A menos que J, D.’ Isso significa que, se você não vestir um casaco, vai pegar um resfriado; com isso em mente, podemos traduzi-la como $\neg J \rightarrow D$. Também significa que, se você não pegar um resfriado, então deve ter vestido um casaco; com isso em mente, podemos traduzi-la como $\neg D \rightarrow J$.

Qual dessas é a tradução correta da sentença 27? As duas traduções são corretas, porque são logicamente equivalentes em SL.

A sentença 28, em português, é logicamente equivalente à 27. Ela também pode ser traduzida tanto como $\neg J \rightarrow D$ quanto como $\neg D \rightarrow J$.

Ao simbolizar sentenças como 27 e 28, é fácil se confundir. Como o condicional não é simétrico, seria errado traduzir qualquer uma delas como $J \rightarrow \neg D$. Felizmente, há outras expressões logicamente equivalentes. Ambas as sentenças significam que você vai vestir um casaco ou se não vestir um casaco então vai pegar um resfriado. Assim, podemos traduzi-las como $J \vee D$. (Você poderia achar que o ‘ou’ aqui deveria ser exclusivo. No entanto, as sentenças não excluem a possibilidade de que você *vista* um casaco *e ainda assim* pegue um resfriado; casacos não protegem contra todas as formas possíveis de pegar um resfriado.)

> Se uma sentença puder ser parafraseada como ‘A menos que $\mathcal{A}$, $\mathcal{B}$’, então ela pode ser simbolizada como $\mathcal{A} \vee \mathcal{B}$.

A simbolização de tipos padrão de sentença é resumida na p. 152.

2.4 Sentenças de SL

A sentença ‘Maçãs são vermelhas, ou frutas vermelhas são azuis’ é uma sentença em português, e a expressão ‘$(A \vee B)$’ é uma sentença de SL. Embora consigamos reconhecer sentenças do português quando as vemos, não temos uma definição formal de 'sentença do português'. Em SL, é possível definir formalmente o que conta como sentença. Esse é um dos aspectos em que uma linguagem formal como SL é mais precisa que uma linguagem natural como o português (ou o inglês).

É importante distinguir entre a linguagem lógica SL, que estamos desenvolvendo, e a linguagem que usamos para falar sobre SL. Quando falamos sobre uma linguagem, a linguagem *de que estamos falando* é chamada de LINGUAGEM-OBJETO. A linguagem que usamos para falar sobre a linguagem-objeto é chamada de METALINGUAGEM.

A linguagem-objeto neste capítulo é SL. A metalinguagem é o inglês matemático aqui vertido para o português, mas ainda suplementado com vocabulário lógico e matemático. A expressão '$(A \vee B)$' é uma sentença na linguagem-objeto, porque usa apenas símbolos de SL. Já a palavra 'sentença' não faz parte de SL; assim, a frase 'Esta expressão é uma sentença de SL' não é uma sentença de SL. É uma sentença na metalinguagem, usada para falar *sobre* SL.

Nesta seção, daremos uma definição formal de 'sentença de SL'. A definição será dada em inglês matemático (metalinguagem), aqui traduzido para o português.

Expressões

Há três tipos de símbolos em SL:

letras de sentença com subscritos, se necessário	$A, B, C, \ldots, Z$ $A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
conectivos	$\neg, \&, \vee, \rightarrow, \leftrightarrow$
parênteses	(,)

Definimos uma EXPRESSÃO DE SL como qualquer sequência (string) de símbolos de SL. Pegue quaisquer símbolos de SL e escreva-os em alguma ordem: você terá uma expressão.

Fórmulas bem formadas

Como qualquer sequência de símbolos é uma expressão, muitas expressões de SL serão simplesmente sem sentido. Uma expressão significativa é chamada de *fórmula bem formada*. É comum usar a sigla em inglês *wff* (well-formed formula); o plural é wffs.

Obviamente, letras de sentença individuais como A e G_{13} serão wffs. Podemos formar novas wffs a partir delas usando os conectivos. Usando negação, obtemos $\neg A$ e $\neg G_{13}$. Usando conjunção, obtemos $A \& G_{13}$, $G_{13} \& A$,

$A \& A$ e $G_{13} \& G_{13}$. Também poderíamos aplicar negação repetidamente e obter wffs como $\neg\neg A$, ou aplicar negação junto com conjunção e obter wffs como $\neg(A \& G_{13})$ e $\neg(G_{13} \& \neg G_{13})$. As combinações possíveis são infinitas, mesmo começando apenas com essas duas letras de sentença, e há infinitas letras de sentença. Portanto, não faz sentido tentar listar todas as wffs.

Em vez disso, descreveremos o processo pelo qual as wffs podem ser construídas. Considere a negação: dada qualquer wff $\mathcal{A}$ de SL, $\neg\mathcal{A}$ é uma wff de SL. É importante notar que $\mathcal{A}$ aqui não é a letra de sentença A. Ela é uma variável que representa qualquer wff. Observe que essa variável $\mathcal{A}$ não é um símbolo de SL, de modo que $\neg\mathcal{A}$ não é uma expressão de SL. Em vez disso, é uma expressão da metalinguagem que nos permite falar sobre infinitas expressões de SL: todas as expressões que começam com o símbolo de negação. Como $\mathcal{A}$ faz parte da metalinguagem, é chamada de *metavariável*.

Podemos dizer algo semelhante para cada um dos outros conectivos. Por exemplo, se $\mathcal{A}$ e $\mathcal{B}$ são wffs de SL, então $(\mathcal{A} \& \mathcal{B})$ é uma wff de SL. Fornecendo cláusulas desse tipo para todos os conectivos, chegamos à seguinte definição formal de fórmula bem formada de SL:

1. Toda sentença atômica é uma wff.
2. Se $\mathcal{A}$ é uma wff, então $\neg\mathcal{A}$ é uma wff de SL.
3. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A} \& \mathcal{B})$ é uma wff.
4. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A} \vee \mathcal{B})$ é uma wff.
5. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A} \rightarrow \mathcal{B})$ é uma wff.
6. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A} \leftrightarrow \mathcal{B})$ é uma wff.
7. Todas e somente as wffs de SL podem ser geradas por aplicações dessas regras.

Note que não podemos aplicar imediatamente essa definição para verificar se uma expressão qualquer é ou não uma wff. Suponha que queiramos saber se $\neg\neg\neg D$ é uma wff de SL. Olhando a segunda cláusula da definição, sabemos que $\neg\neg\neg D$ é uma wff *se* $\neg\neg D$ for uma wff. Então precisamos perguntar se $\neg\neg D$ é uma wff. De novo, pela segunda cláusula, $\neg\neg D$ é uma wff *se* $\neg D$ for. E, por sua vez, $\neg D$ é uma wff *se* D for uma wff. Agora, D é uma letra de sentença, uma sentença atômica de SL, então sabemos pela primeira cláusula que D é uma wff. Assim, para uma fórmula composta como $\neg\neg\neg D$, precisamos aplicar a definição repetidamente. Eventualmente, chegamos às sentenças atômicas a partir das quais a wff é construída.

Definições desse tipo são chamadas de *recursivas*. Definições recursivas começam com alguns elementos-base especificáveis e definem maneiras de compor indefinidamente esses elementos-base. Assim como a definição recursiva permite construir sentenças complexas a partir de partes simples, podemos usá-la para decompor sentenças em partes mais simples. Para determinar se algo se encaixa na definição, podemos precisar recorrer a ela muitas vezes.

O conectivo que você observa primeiro ao decompor uma sentença é chamado de PRINCIPAL OPERADOR LÓGICO (ou operador lógico principal) daquela sentença. Por exemplo: o operador lógico principal de $\neg(E \vee (F \rightarrow G))$ é a negação, $\neg$. O operador lógico principal de $(\neg E \vee (F \rightarrow G))$ é a disjunção, $\vee$.

Sentenças

Lembre-se de que uma sentença é uma expressão significativa que pode ser verdadeira ou falsa. Como as expressões significativas de SL são as wffs e como toda wff de SL é verdadeira ou falsa, a definição de sentença de SL coincide com a definição de wff. Nem toda linguagem formal terá essa propriedade agradável. Na linguagem QL, desenvolvida mais adiante no livro, há wffs que não são sentenças.

A estrutura recursiva das sentenças em SL será importante quando considerarmos as circunstâncias em que uma sentença é verdadeira ou falsa. A sentença $\neg\neg\neg D$ é verdadeira se, e somente se, a sentença $\neg\neg D$ for falsa; e assim por diante, ao longo da estrutura, até chegarmos ao componente atômico: $\neg\neg\neg D$ é verdadeira se, e somente se, a sentença atômica D for falsa. Voltaremos a esse ponto no próximo capítulo.

Convenções notacionais

Uma wff como $(Q \,\&\, R)$ precisa estar cercada por parênteses, porque poderíamos aplicar de novo a definição e usar isso como parte de uma sentença ainda mais complicada. Se negarmos $(Q \,\&\, R)$, obtemos $\neg(Q \,\&\, R)$. Se nós tivéssemos apenas $Q \,\&\, R$ sem parênteses e colocássemos uma negação na frente, teríamos $\neg Q \,\&\, R$. É mais natural ler isso como significando o mesmo que $(\neg Q \,\&\, R)$, algo muito diferente de $\neg(Q \,\&\, R)$. A sentença $\neg(Q \,\&\, R)$ diz que não é o caso que tanto Q quanto R sejam verdadeiros; Q pode ser falso, ou R pode ser falso, mas a sentença não nos diz qual. Já $(\neg Q \,\&\, R)$ diz especificamente que Q é falso e R é verdadeiro. Assim, os parênteses são cruciais para o significado.

Portanto, estritamente falando, $Q \,\&\, R$ sem parênteses *não* é uma sentença de SL. No uso prático de SL, porém, muitas vezes poderemos relaxar a definição precisa para facilitar nossa vida. Faremos isso de várias maneiras.

Primeiro, entendemos que $Q \,\&\, R$ significa o mesmo que $(Q \,\&\, R)$. Por convenção, podemos omitir parênteses que ocorram *em torno de toda a sentença.*

Segundo, sentenças longas com muitos pares de parênteses encaixados podem ser difíceis de ler. Adotamos a convenção de usar colchetes '[' e ']' no lugar dos parênteses. Não há diferença lógica entre $(P \vee Q)$ e $[P \vee Q]$, por exemplo. A sentença pesada

$$(((H \to I) \vee (I \to H)) \,\&\, (J \vee K))$$

poderia ser escrita assim:

$$\big[(H \to I) \vee (I \to H)\big] \,\&\, (J \vee K)$$

Terceiro, às vezes queremos traduzir a conjunção de três ou mais sentenças. Para a sentença 'Alice, Bob e Candice foram todos à festa', suponha que A signifique 'Alice foi', B 'Bob foi' e C 'Candice foi'. A definição só nos permite formar conjunção de duas sentenças de cada vez, então podemos traduzir como $(A \,\&\, B) \,\&\, C$ ou como $A \,\&\, (B \,\&\, C)$. Não há motivo para distinguir essas opções, já que são logicamente equivalentes. Não há diferença lógica entre a primeira, em que $(A \,\&\, B)$ é conjuntada com C, e a segunda, em que A é conjuntada com $(B \,\&\, C)$. Portanto, podemos simplesmente escrever $A \,\&\, B \,\&\, C$. Por convenção, podemos omitir parênteses quando conjuntamos três ou mais sentenças.

Quarto, uma situação semelhante ocorre com múltiplas disjunções. 'Ou Alice, Bob ou Candice foi à festa' pode ser traduzida como $(A \vee B) \vee C$ ou como $A \vee (B \vee C)$. Como as duas traduções são logicamente equivalentes, podemos escrever $A \vee B \vee C$.

Essas duas últimas convenções só valem para múltiplas conjunções ou múltiplas disjunções. Se uma série de conectivos inclui tanto disjunções quanto conjunções, então os parênteses são essenciais; como em $(A \,\&\, B) \vee C$ e $A \,\&\, (B \vee C)$. Os parênteses também são necessários se há uma série de condicionais ou bicondicionais; como em $(A \to B) \to C$ e $A \leftrightarrow (B \leftrightarrow C)$.

Adotamos essas quatro regras como *convenções notacionais*, e não como mudanças na definição de sentença. Estritamente falando, $A \vee B \vee C$ ainda não é uma sentença. Em vez disso, é um tipo de abreviação. Escrevemo-la por conveniência, mas queremos dizer, na verdade, a sentença $(A \vee (B \vee C))$.

Se tivéssemos dado uma definição diferente de wff, poderíamos fazer com que tais abreviações fossem wffs. Poderíamos ter escrito a regra 3 assim: Se $\mathcal{A}$, $\mathcal{B}$, … $\mathcal{Z}$ são wffs, então $(\mathcal{A} \,\&\, \mathcal{B} \,\&\, \ldots \,\&\, \mathcal{Z})$ é uma wff. Isso tornaria mais fácil traduzir algumas sentenças em português, mas teria o custo de tornar nossa linguagem formal mais complicada. Teríamos de carregar essa definição complexa quando desenvolvêssemos tabelas de verdade e o sistema de provas. Queremos uma linguagem lógica que seja *simples do ponto de vista formal* e ainda assim permita traduzir bem a partir do português (ou do inglês). Adotar convenções notacionais é um meio-termo entre essas duas exigências.

Practice Exercises

$\star$ **Part A** Usando a chave de simbolização dada, traduza cada sentença em português para SL.

M: Aqueles seres são homens fantasiados.
C: Aqueles seres são chimpanzés.
G: Aqueles seres são gorilas.

1. Aqueles seres não são homens fantasiados.
2. Aqueles seres são homens fantasiados, ou não são.
3. Aqueles seres são gorilas ou são chimpanzés.
4. Aqueles seres não são nem gorilas nem chimpanzés.
5. Se aqueles seres são chimpanzés, então eles não são nem gorilas nem homens fantasiados.
6. A menos que aqueles seres sejam homens fantasiados, eles são ou chimpanzés ou gorilas.

Part B Usando a chave de simbolização dada, traduza cada sentença em português para SL.

A: Mister Ace foi assassinado.
B: O mordomo fez isso.
C: A cozinheira fez isso.
D: A Duquesa está mentindo.
E: Mister Edge foi assassinado.
F: A arma do crime foi uma frigideira.

1. Ou Mister Ace ou Mister Edge foi assassinado.
2. Se Mister Ace foi assassinado, então a cozinheira fez isso.
3. Se Mister Edge foi assassinado, então a cozinheira não fez isso.
4. Ou o mordomo fez isso, ou a Duquesa está mentindo.
5. A cozinheira fez isso somente se a Duquesa estiver mentindo.
6. Se a arma do crime foi uma frigideira, então a culpada só pode ter sido a cozinheira.
7. Se a arma do crime não foi uma frigideira, então a culpada foi ou a cozinheira ou o mordomo.
8. Mister Ace foi assassinado se e somente se Mister Edge não foi assassinado.
9. A Duquesa está mentindo, a menos que tenha sido Mister Edge o assassinado.
10. Se Mister Ace foi assassinado, ele foi morto com uma frigideira.
11. Já que a cozinheira fez isso, o mordomo não fez.
12. É claro que a Duquesa está mentindo!

⋆ **Part C** Usando a chave de simbolização dada, traduza cada sentença em português para SL.

$\mathbf{E_1}$**:** Ava é eletricista.
$\mathbf{E_2}$**:** Harrison é eletricista.
$\mathbf{F_1}$**:** Ava é bombeira.
$\mathbf{F_2}$**:** Harrison é bombeiro.

S_1: Ava está satisfeita com sua carreira.
S_2: Harrison está satisfeito com sua carreira.

1. Ava e Harrison são ambos eletricistas.
2. Se Ava é bombeira, então ela está satisfeita com sua carreira.
3. Ava é bombeira, a menos que seja eletricista.
4. Harrison é um eletricista insatisfeito.
5. Nem Ava nem Harrison é eletricista.
6. Ava e Harrison são ambos eletricistas, mas nenhum dos dois acha isso satisfatório.
7. Harrison está satisfeito somente se ele for bombeiro.
8. Se Ava não é eletricista, então Harrison também não é, mas se ela é, então ele também é.
9. Ava está satisfeita com sua carreira se e somente se Harrison não estiver satisfeito com a dele.
10. Se Harrison é simultaneamente eletricista e bombeiro, então ele deve estar satisfeito com o trabalho.
11. Não pode ser que Harrison seja ao mesmo tempo eletricista e bombeiro.
12. Harrison e Ava são ambos bombeiros se e somente se nenhum dos dois for eletricista.

⋆ **Part D** Dê uma chave de simbolização e simbolize as sentenças a seguir em SL.

1. Alice e Bob são ambos espiões.
2. Se ou Alice ou Bob é espião, então o código foi decifrado.
3. Se nem Alice nem Bob é espião, então o código permanece indecifrado.
4. A embaixada alemã ficará em alvoroço, a menos que alguém tenha decifrado o código.
5. Ou o código foi decifrado ou não foi, mas a embaixada alemã ficará em alvoroço de qualquer forma.
6. Ou Alice ou Bob é espião, mas não ambos.

Part E Dê uma chave de simbolização e simbolize as sentenças a seguir em SL.

1. Se Gregor jogar na primeira base, então o time vai perder.
2. O time vai perder, a menos que aconteça um milagre.
3. O time ou vai perder ou não vai, mas Gregor vai jogar na primeira base de qualquer forma.
4. A mãe de Gregor vai assar biscoitos se e somente se Gregor jogar na primeira base.
5. Se acontecer um milagre, então a mãe de Gregor não vai assar biscoitos.

Part F Para cada argumento, escreva uma chave de simbolização e traduza o argumento o melhor possível em SL.

1. Se Dorothy toca piano de manhã, então Roger acorda mal-humorado. Dorothy toca piano de manhã, a menos que esteja distraída. Logo, se Roger não acorda mal-humorado, então Dorothy deve estar distraída.
2. Ou vai chover ou vai nevar na terça-feira. Se chover, Neville ficará triste. Se nevar, Neville sentirá frio. Portanto, Neville ficará ou triste ou com frio na terça-feira.
3. Se Zoog lembrou de fazer os afazeres, então as coisas estão limpas, mas não arrumadas. Se ele se esqueceu, então as coisas estão arrumadas, mas não limpas. Portanto, as coisas estão ou arrumadas ou limpas mas não ambas.

⋆ **Part G** Para cada expressão a seguir: (a) Ela é uma wff de SL? (b) Ela é uma sentença de SL, levando em conta as convenções notacionais?

1. (A)
2. $J_{374} \vee \neg J_{374}$
3. $\neg\neg\neg\neg F$
4. $\neg \,\&\, S$
5. $(G \,\&\, \neg G)$
6. $\mathcal{A} \rightarrow \mathcal{A}$
7. $(A \rightarrow (A \,\&\, \neg F)) \vee (D \leftrightarrow E)$
8. $[(Z \leftrightarrow S) \rightarrow W] \,\&\, [J \vee X]$
9. $(F \leftrightarrow \neg D \rightarrow J) \vee (C \,\&\, D)$

Part H

1. Existe alguma wff de SL que não contenha letras de sentença? Por quê?
2. No capítulo, simbolizamos um *ou exclusivo* usando $\vee$, $\&$ e $\neg$. Como você poderia traduzir um *ou exclusivo* usando apenas dois conectivos? Há alguma maneira de traduzir um *ou exclusivo* usando apenas um conectivo?

Chapter 3

Tabelas-verdade

Este capítulo introduz um modo de avaliar sentenças e argumentos de SL. Embora possa ser trabalhoso, o método das tabelas-verdade é um procedimento puramente mecânico, que não exige intuição nem qualquer tipo de insight especial.

3.1 Conectivos verifuncionais

Qualquer sentença não atômica de SL é composta de sentenças atômicas com conectivos sentenciais. O valor de verdade da sentença composta depende apenas dos valores de verdade das sentenças atômicas que a compõem. Para saber o valor de verdade de $(D \leftrightarrow E)$, por exemplo, você só precisa saber o valor de verdade de D e o valor de verdade de E. Conectivos que funcionam desse modo são chamados de VERIFUNCIONAIS (isto é, seu comportamento depende apenas dos valores de verdade das partes).

Neste capítulo, exploraremos o fato de que todos os operadores lógicos de SL são verifuncionais isso torna possível construir tabelas-verdade para determinar as propriedades lógicas das sentenças. É importante perceber, porém, que isso não vale para todas as linguagens. Em português (como em inglês), é possível formar uma nova sentença a partir de uma sentença mais simples $\mathcal{X}$ dizendo: É possível que $\mathcal{X}$. O valor de verdade dessa nova sentença não depende diretamente do valor de verdade de $\mathcal{X}$. Mesmo que $\mathcal{X}$ seja falsa, pode ser que, em algum sentido, $\mathcal{X}$ *pudesse* ter sido verdadeira nesse caso, a nova sentença seria verdadeira. Alguns sistemas formais, chamados de *lógicas modais*, têm um operador para possibilidade. Em uma lógica modal, poderíamos traduzir É possível que $\mathcal{X}$ como $\Diamond\mathcal{X}$. No entanto, a possibilidade de traduzir sentenças como essa vem com um custo: o operador $\Diamond$ não é verifuncional, e por isso lógicas modais não são adequadas para o uso de tabelas-verdade da forma simples que veremos aqui.

3.2 Tabelas-verdade completas

O valor de verdade de sentenças que contêm apenas um conectivo é dado pela tabela-verdade característica desse conectivo. No capítulo anterior, escrevemos essas tabelas-verdade com 'T' para verdadeiro e 'F' para falso. É importante notar, porém, que aqui não estamos falando de verdade em nenhum sentido profundo ou cosmológico. Poetas e filósofos podem discutir longamente sobre a natureza e o significado de *verdade*, mas as funções de verdade em SL são apenas regras que transformam valores de entrada em valores de saída. Para destacar isso, neste capítulo escreveremos '1' e '0' em vez de 'T' e 'F'. Mesmo que interpretemos '1' como verdadeiro e '0' como falso, computadores podem ser programados para preencher tabelas-verdade de modo puramente mecânico. Em uma máquina, '1' pode significar que um registrador está ligado, e '0', que o registrador está desligado. Do ponto de vista matemático, eles são apenas os dois valores possíveis que uma sentença de SL pode assumir.

Aqui estão as tabelas-verdade dos conectivos de SL, escritas em termos de 1s e 0s.

$\mathcal{A}$	$\neg\mathcal{A}$
1	0
0	1

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\,\&\,\mathcal{B}$	$\mathcal{A}\lor\mathcal{B}$	$\mathcal{A}\to\mathcal{B}$	$\mathcal{A}\leftrightarrow\mathcal{B}$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

Tabela 3.1: As tabelas-verdade características dos conectivos de SL.

A tabela-verdade característica da conjunção, por exemplo, fornece as condições de verdade de qualquer sentença da forma $(\mathcal{A}\,\&\,\mathcal{B})$. Mesmo que os conjunos $\mathcal{A}$ e $\mathcal{B}$ sejam sentenças longas e complicadas, a conjunção é verdadeira se, e somente se, tanto $\mathcal{A}$ quanto $\mathcal{B}$ forem verdadeiras. Considere a sentença $(H\,\&\,I) \to H$. Tomamos todas as possíveis combinações de verdadeiro e falso para H e I, o que nos dá quatro linhas. Copiamos então os valores de verdade das letras sentenciais e os escrevemos sob as letras na sentença.

H	I	$(H$	$\&$	$I)$	$\to$	H
1	1	**1**		**1**		**1**
1	0	**1**		**0**		**1**
0	1	**0**		**1**		**0**
0	0	**0**		**0**		**0**

Agora considere a subsentença $H\,\&\,I$. Ela é uma conjunção $\mathcal{A}\,\&\,\mathcal{B}$ em que H faz o papel de $\mathcal{A}$ e I faz o papel de $\mathcal{B}$. H e I são ambas verdadeiras na primeira linha. Como uma conjunção é verdadeira quando ambos os conjunos são verdadeiros, escrevemos 1 sob o símbolo de conjunção. Continuamos para as outras três linhas e obtemos:

H	I	$(H$	$\&$	$I)$	$\to$	H
		$\mathcal{A}$	$\&$	$\mathcal{B}$		
1	1	1	**1**	1		1
1	0	1	**0**	0		1
0	1	0	**0**	1		0
0	0	0	**0**	0		0

A sentença inteira é um condicional $\mathcal{A} \to \mathcal{B}$, em que $(H \& I)$ é $\mathcal{A}$ e H é $\mathcal{B}$. Na segunda linha, por exemplo, $(H \& I)$ é falsa e H é verdadeira. Como um condicional é verdadeiro quando o antecedente é falso, escrevemos 1 na segunda linha sob o símbolo do condicional. Fazemos isso para as demais linhas e obtemos:

H	I	$(H$	$\&$	$I)$	$\to$	H
			$\mathcal{A}$		$\to$	$\mathcal{B}$
1	1		1		**1**	1
1	0		0		**1**	1
0	1		0		**1**	0
0	0		0		**1**	0

A coluna de 1s sob o condicional nos mostra que a sentença $(H \& I) \to I$ é verdadeira independentemente dos valores de verdade de H e I. Eles podem ser verdadeiros ou falsos em qualquer combinação e, ainda assim, a sentença composta sai verdadeira. É crucial que tenhamos considerado todas as combinações possíveis. Se tivéssemos apenas uma tabela de duas linhas, não poderíamos ter certeza de que a sentença não seria falsa em alguma outra combinação de valores de verdade.

Neste exemplo, não repetimos todas as entradas em cada tabela sucessiva. Mas, quando realmente escrevemos tabelas-verdade no papel, é impraticável apagar colunas inteiras ou reescrever a tabela toda a cada passo. Embora fique mais apertado, a tabela-verdade pode ser escrita assim:

H	I	$(H$	$\&$	$I)$	$\to$	H
1	1	1	1	1	1	1
1	0	1	0	0	1	1
0	1	0	0	1	1	0
0	0	0	0	0	1	0

A maior parte das colunas sob a sentença está ali apenas para controle de contas. Quando você ficar mais à vontade com tabelas-verdade, provavelmente não precisará mais copiar as colunas de cada letra sentencial. De todo modo, o valor de verdade da sentença em cada linha é apenas o valor na coluna sob o conectivo lógico principal da sentença; neste caso, a coluna sob o condicional.

Uma TABELA-VERDADE COMPLETA tem uma linha para cada combinação possível de 1 e 0 para todas as letras sentenciais envolvidas. O tamanho da tabela-verdade completa depende do número de letras sentenciais diferentes na sentença. Uma sentença que contém apenas uma letra sentencial exige apenas duas linhas, como na tabela característica da negação. Isso continua valendo mesmo que a mesma letra apareça muitas vezes, como na sentença $[(C \leftrightarrow C) \rightarrow C] \,\&\, \neg(C \rightarrow C)$. A tabela-verdade completa exige apenas duas linhas porque há apenas duas possibilidades: C pode ser verdadeira ou pode ser falsa. Uma mesma letra sentencial nunca pode ser marcada ao mesmo tempo como 1 e como 0 na mesma linha. A tabela-verdade para essa sentença fica assim:

C	$[(C$	$\leftrightarrow$	$C)$	$\rightarrow$	$C]$	$\&$	$\neg$	$(C$	$\rightarrow$	$C)$
1	1	1	1	1	1	**0**	0	1	1	1
0	0	1	0	0	0	**0**	0	0	1	0

Observando a coluna sob o conectivo principal, vemos que a sentença é falsa em ambas as linhas da tabela; isto é, ela é falsa tanto se C é verdadeira quanto se C é falsa.

Uma sentença que contém duas letras sentenciais exige quatro linhas para uma tabela-verdade completa, como nas tabelas características e na tabela de $(H \,\&\, I) \rightarrow I$.

Uma sentença que contém três letras sentenciais exige oito linhas. Por exemplo:

M	N	P	M	$\&$	$(N$	$\vee$	$P)$
1	1	1	1	**1**	1	1	1
1	1	0	1	**1**	1	1	0
1	0	1	1	**1**	0	1	1
1	0	0	1	**0**	0	0	0
0	1	1	0	**0**	1	1	1
0	1	0	0	**0**	1	1	0
0	0	1	0	**0**	0	1	1
0	0	0	0	**0**	0	0	0

A partir dessa tabela, sabemos que a sentença $M \,\&\, (N \vee P)$ pode ser verdadeira ou falsa dependendo dos valores de verdade de M, N e P.

Uma tabela-verdade completa para uma sentença com quatro letras sentenciais diferentes exige 16 linhas. Com cinco letras, 32 linhas. Com seis letras, 64 linhas. E assim por diante. Em geral: se uma tabela-verdade completa tem n letras sentenciais diferentes, então ela deve ter 2^n linhas.

Para preencher as colunas de uma tabela-verdade completa, comece pela letra sentencial mais à direita e alterne 1s e 0s. Na próxima coluna à esquerda, escreva dois 1s, depois dois 0s, e repita. Para a terceira letra, escreva quatro 1s seguidos de quatro 0s. Isso gera uma tabela de oito linhas, como acima. Para uma tabela de 16 linhas, a próxima coluna de letras sentenciais deve ter oito 1s seguidos de oito 0s. Para uma tabela de 32 linhas, a próxima coluna terá 16 1s seguidos de 16 0s. E assim por diante.

3.3 Usando tabelas-verdade

Tautologias, contradições e sentenças contingentes

Recorde que uma sentença (em linguagem natural) é uma tautologia se ela tem de ser verdadeira como questão de lógica. Com uma tabela-verdade completa, consideramos todos os modos como o mundo poderia ser. Se a sentença é verdadeira em todas as linhas de uma tabela-verdade completa, então ela é verdadeira como questão de lógica, independentemente de como o mundo de fato é.

Assim, uma sentença é uma TAUTOLOGIA EM SL se a coluna sob seu conectivo principal tem 1 em todas as linhas de uma tabela-verdade completa.

De modo análogo, uma sentença é uma CONTRADIÇÃO EM SL se a coluna sob seu conectivo principal tem 0 em todas as linhas de uma tabela-verdade completa.

Uma sentença é CONTINGENTE EM SL se ela não é nem tautologia nem contradição; isto é, se ela tem 1 em pelo menos uma linha e 0 em pelo menos uma outra linha.

Pelas tabelas da seção anterior, sabemos que $(H \,\&\, I) \rightarrow H$ é uma tautologia, que $[(C \leftrightarrow C) \rightarrow C] \,\&\, \neg(C \rightarrow C)$ é uma contradição, e que $M \,\&\, (N \vee P)$ é contingente.

Equivalência lógica

Duas sentenças são logicamente equivalentes em português se elas têm o mesmo valor de verdade como questão de lógica. Mais uma vez, as tabelas-verdade nos permitem definir um conceito análogo para SL: duas sentenças são LOGICAMENTE EQUIVALENTES EM SL se elas têm o mesmo valor de verdade em todas as linhas de uma tabela-verdade completa.

Considere as sentenças $\neg(A \vee B)$ e $\neg A \,\&\, \neg B$. Elas são logicamente equivalentes? Para descobrir, construímos uma tabela-verdade:

A	B	$\neg\,(A \vee B)$	$\neg\, A \,\&\, \neg\, B$
1	1	**0** 1 1 1	0 1 **0** 0 1
1	0	**0** 1 1 0	0 1 **0** 1 0
0	1	**0** 0 1 1	1 0 **0** 0 1
0	0	**1** 0 0 0	1 0 **1** 1 0

Observe as colunas dos conectivos principais: negação, na primeira sentença, e conjunção, na segunda. Nas três primeiras linhas, ambas têm valor 0. Na última linha, ambas têm valor 1. Como coincidem em todas as linhas, as duas sentenças são logicamente equivalentes.

Consistência

Um conjunto de sentenças em português é consistente se é logicamente possível que todas sejam verdadeiras ao mesmo tempo. Um conjunto de sentenças é LOGICAMENTE CONSISTENTE EM SL se existe pelo menos uma linha de uma tabela-verdade completa em que todas as sentenças sejam verdadeiras (todas com valor 1). Ele é INCONSISTENTE caso contrário (isto é, se não há linha em que todas sejam verdadeiras).

Validade

Um argumento em português é válido se é logicamente impossível que as premissas sejam verdadeiras e a conclusão falsa ao mesmo tempo. Um argumento é VÁLIDO EM SL se não há linha de uma tabela-verdade completa em que todas as premissas sejam 1 e a conclusão seja 0; o argumento é INVÁLIDO EM SL se existe uma linha assim.

Considere este argumento:

$$\begin{array}{rl} & \neg L \to (J \vee L) \\ & \neg L \\ \therefore & J \end{array}$$

Ele é válido? Para descobrir, construímos uma tabela-verdade.

J	L	$\neg\, L \to (J \vee L)$	$\neg$ L	J
1	1	0 1 **1** 1 1 1	**0** 1	**1**
1	0	1 0 **1** 1 1 0	**1** 0	**1**
0	1	0 1 **1** 0 1 1	**0** 1	**0**
0	0	1 0 **0** 0 0 0	**1** 0	**0**

Sim, o argumento é válido. A única linha em que ambas as premissas são 1 é a segunda linha, e nessa linha a conclusão também é 1.

3.4 Tabelas-verdade parciais

Para mostrar que uma sentença é uma tautologia, precisamos mostrar que ela vale 1 em todas as linhas. Portanto, precisamos de uma tabela-verdade completa. Mas, para mostrar que uma sentença *não* é uma tautologia, basta uma linha: uma linha em que a sentença tenha valor 0. Assim, para mostrar que algo não é tautologia, basta fornecer uma *tabela-verdade parcial* de uma linha não importa quantas letras sentenciais a sentença contenha.

Considere, por exemplo, a sentença $(U \& T) \to (S \& W)$. Queremos mostrar que ela *não* é uma tautologia, fornecendo uma tabela-verdade parcial. Preenchemos a coluna da sentença inteira com 0. O conectivo principal da sentença é um condicional. Para que o condicional seja falso, o antecedente deve ser verdadeiro (1) e o consequente, falso (0). Assim, preenchemos:

S	T	U	W	$(U$	$\&$	$T)$	$\to$	$(S$	$\&$	$W)$
					1		**0**		0	

Para que $(U \& T)$ seja verdadeira, tanto U quanto T devem ser verdadeiras.

S	T	U	W	$(U$	$\&$	$T)$	$\to$	$(S$	$\&$	$W)$
	1	1		1	1	1	**0**		0	

Agora só precisamos tornar $(S \& W)$ falsa. Para isso, basta tornar pelo menos uma das sentenças S ou W falsa. Podemos tornar as duas falsas, se quisermos. O importante é que a sentença inteira saia falsa nessa linha. Fazendo uma escolha arbitrária, completamos a tabela assim:

S	T	U	W	$(U$	$\&$	$T)$	$\to$	$(S$	$\&$	$W)$
0	1	1	0	1	1	1	**0**	0	0	0

Mostrar que algo é uma contradição exige uma tabela-verdade completa. Mostrar que algo *não* é uma contradição exige apenas uma tabela-verdade parcial de uma linha, em que a sentença seja verdadeira.

Uma sentença é contingente se não é nem tautologia nem contradição. Então, mostrar que uma sentença é contingente exige uma tabela-verdade parcial de *duas linhas*: a sentença deve ser verdadeira em uma linha e falsa em outra. Por exemplo, podemos mostrar que a sentença acima é contingente com a seguinte tabela:

S	T	U	W	$(U$	$\&$	$T)$	$\to$	$(S$	$\&$	$W)$
0	1	1	0	1	1	1	**0**	0	0	0
0	1	0	0	0	0	1	**1**	0	0	0

Note que há muitas combinações de valores de verdade que tornariam a sentença verdadeira, portanto há muitas maneiras possíveis de escrever a segunda linha.

Mostrar que uma sentença *não* é contingente exige fornecer uma tabela-verdade completa, porque isso requer mostrar que a sentença é uma tautologia ou uma contradição. Se você não sabe se uma sentença é contingente, então não sabe de antemão se será necessária uma tabela completa ou parcial. Você sempre pode começar construindo a tabela completa. Se, no caminho, você encontrar linhas que mostram que a sentença é contingente, pode parar. Caso contrário, conclua a tabela. Embora duas linhas bem escolhidas sejam suficientes para mostrar que uma sentença contingente é contingente, não há nada de errado em preencher mais linhas.

Mostrar que duas sentenças são logicamente equivalentes requer uma tabela-verdade completa. Mostrar que duas sentenças *não* são logicamente equivalentes requer apenas uma tabela-verdade parcial de uma linha: basta construí-la de modo que uma das sentenças seja verdadeira e a outra falsa.

Mostrar que um conjunto de sentenças é consistente requer fornecer uma linha de uma tabela-verdade em que todas as sentenças sejam verdadeiras. O resto da tabela é irrelevante, então uma tabela parcial de uma linha basta. Mostrar que um conjunto de sentenças é inconsistente, por outro lado, exige uma tabela-verdade completa: é preciso mostrar que, em toda linha, pelo menos uma das sentenças é falsa.

Mostrar que um argumento é válido requer uma tabela-verdade completa. Mostrar que um argumento é *inválido* exige apenas fornecer uma tabela-verdade de uma linha: se você consegue produzir uma linha em que as premissas são todas verdadeiras e a conclusão é falsa, então o argumento é inválido.

A tabela a seguir resume quando é necessária uma tabela-verdade completa e quando uma tabela parcial é suficiente.

	SIM	NÃO
tautologia?	tabela-verdade completa	tabela-verdade parcial de uma linha
contradição?	tabela-verdade completa	tabela-verdade parcial de uma linha
contingente?	tabela-verdade parcial de duas linhas	tabela-verdade completa
equivalente?	tabela-verdade completa	tabela-verdade parcial de uma linha
consistente?	tabela-verdade parcial de uma linha	tabela-verdade completa
válido?	tabela-verdade completa	tabela-verdade parcial de uma linha

Tabela 3.2: Você precisa de uma tabela-verdade completa ou parcial? Depende do que você quer mostrar.

Practice Exercises

Se você quiser mais prática, pode construir tabelas-verdade para quaisquer sentenças e argumentos dos exercícios do capítulo anterior.

⋆ **Part A** Determine se cada sentença é uma tautologia, uma contradição ou uma sentença contingente. Justifique sua resposta com uma tabela-verdade completa ou parcial, conforme apropriado.

1. $A \to A$
2. $\neg B \,\&\, B$
3. $C \to \neg C$
4. $\neg D \vee D$
5. $(A \leftrightarrow B) \leftrightarrow \neg(A \leftrightarrow \neg B)$
6. $(A \,\&\, B) \vee (B \,\&\, A)$
7. $(A \to B) \vee (B \to A)$
8. $\neg[A \to (B \to A)]$
9. $(A \,\&\, B) \to (B \vee A)$
10. $A \leftrightarrow [A \to (B \,\&\, \neg B)]$
11. $\neg(A \vee B) \leftrightarrow (\neg A \,\&\, \neg B)$
12. $\neg(A \,\&\, B) \leftrightarrow A$
13. $\big[(A \,\&\, B) \,\&\, \neg(A \,\&\, B)\big] \,\&\, C$
14. $A \to (B \vee C)$
15. $[(A \,\&\, B) \,\&\, C] \to B$
16. $(A \,\&\, \neg A) \to (B \vee C)$
17. $\neg\big[(C \vee A) \vee B\big]$
18. $(B \,\&\, D) \leftrightarrow [A \leftrightarrow (A \vee C)]$

⋆ **Part B** Determine se cada par de sentenças é logicamente equivalente. Justifique sua resposta com uma tabela-verdade completa ou parcial, conforme apropriado.

1. A, $\neg A$
2. A, $A \vee A$
3. $A \to A$, $A \leftrightarrow A$
4. $A \vee \neg B$, $A \to B$
5. $A \,\&\, \neg A$, $\neg B \leftrightarrow B$
6. $\neg(A \,\&\, B)$, $\neg A \vee \neg B$
7. $\neg(A \to B)$, $\neg A \to \neg B$
8. $(A \to B)$, $(\neg B \to \neg A)$
9. $[(A \vee B) \vee C]$, $[A \vee (B \vee C)]$
10. $[(A \vee B) \,\&\, C]$, $[A \vee (B \,\&\, C)]$

⋆ **Part C** Determine se cada conjunto de sentenças é consistente ou inconsistente. Justifique sua resposta com uma tabela-verdade completa ou parcial, conforme apropriado.

1. $A \to A$, $\neg A \to \neg A$, $A \,\&\, A$, $A \vee A$
2. $A \,\&\, B$, $C \to \neg B$, C
3. $A \vee B$, $A \to C$, $B \to C$
4. $A \to B$, $B \to C$, A, $\neg C$
5. $B \,\&\, (C \vee A)$, $A \to B$, $\neg(B \vee C)$
6. $A \vee B$, $B \vee C$, $C \to \neg A$

7. $A \leftrightarrow (B \vee C)$, $C \rightarrow \neg A$, $A \rightarrow \neg B$
8. A, B, C, $\neg D$, $\neg E$, F

⋆ **Part D** Determine se cada argumento é válido ou inválido. Justifique sua resposta com uma tabela-verdade completa ou parcial, conforme apropriado.

1. $A \rightarrow A$, $\therefore A$
2. $A \vee \big[A \rightarrow (A \leftrightarrow A)\big]$, $\therefore$ A
3. $A \rightarrow (A \,\&\, \neg A)$, $\therefore \neg A$
4. $A \leftrightarrow \neg(B \leftrightarrow A)$, $\therefore A$
5. $A \vee (B \rightarrow A)$, $\therefore \neg A \rightarrow \neg B$
6. $A \rightarrow B$, B, $\therefore A$
7. $A \vee B$, $B \vee C$, $\neg A$, $\therefore B \,\&\, C$
8. $A \vee B$, $B \vee C$, $\neg B$, $\therefore A \,\&\, C$
9. $(B \,\&\, A) \rightarrow C$, $(C \,\&\, A) \rightarrow B$, $\therefore (C \,\&\, B) \rightarrow A$
10. $A \leftrightarrow B$, $B \leftrightarrow C$, $\therefore A \leftrightarrow C$

⋆ **Part E** Responda cada questão abaixo e justifique sua resposta.

1. Suponha que $\mathcal{A}$ e $\mathcal{B}$ sejam logicamente equivalentes. O que você pode dizer sobre $\mathcal{A} \leftrightarrow \mathcal{B}$?
2. Suponha que $(\mathcal{A} \,\&\, \mathcal{B}) \rightarrow \mathcal{C}$ seja contingente. O que você pode dizer sobre o argumento $\mathcal{A}$, $\mathcal{B}$, $\therefore \mathcal{C}$?
3. Suponha que $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ seja inconsistente. O que você pode dizer sobre $(\mathcal{A} \,\&\, \mathcal{B} \,\&\, \mathcal{C})$?
4. Suponha que $\mathcal{A}$ seja uma contradição. O que você pode dizer sobre o argumento $\mathcal{A}$, $\mathcal{B}$, $\therefore \mathcal{C}$?
5. Suponha que $\mathcal{C}$ seja uma tautologia. O que você pode dizer sobre o argumento $\mathcal{A}$, $\mathcal{B}$, $\therefore \mathcal{C}$?
6. Suponha que $\mathcal{A}$ e $\mathcal{B}$ sejam logicamente equivalentes. O que você pode dizer sobre $(\mathcal{A} \vee \mathcal{B})$?
7. Suponha que $\mathcal{A}$ e $\mathcal{B}$ *não* sejam logicamente equivalentes. O que você pode dizer sobre $(\mathcal{A} \vee \mathcal{B})$?

Part F Poderíamos eliminar o bicondicional ($\leftrightarrow$) da linguagem. Se fizéssemos isso, ainda poderíamos escrever '$A \leftrightarrow B$' para tornar as sentenças mais fáceis de ler, mas isso seria apenas uma abreviação para $(A \rightarrow B) \,\&\, (B \rightarrow A)$. A linguagem resultante ainda seria formalmente equivalente a SL, já que $A \leftrightarrow B$ e $(A \rightarrow B) \,\&\, (B \rightarrow A)$ são logicamente equivalentes em SL. Se déssemos mais peso à simplicidade formal do que à riqueza expressiva, poderíamos substituir mais conectivos por convenções notacionais e ainda ter uma linguagem equivalente a SL.

Existem várias linguagens equivalentes com apenas dois conectivos. Seria suficiente ter apenas negação e o condicional material. Mostre isso escrevendo sentenças logicamente equivalentes a cada uma das seguintes, usando apenas parênteses, letras sentenciais, negação ($\neg$) e condicional material ($\rightarrow$).

⋆ 1. $A \vee B$
⋆ 2. $A \& B$
⋆ 3. $A \leftrightarrow B$

Podemos ter uma linguagem equivalente a SL com apenas negação e disjunção como conectivos. Mostre isso: usando apenas parênteses, letras sentenciais, negação ($\neg$) e disjunção ($\vee$), escreva sentenças logicamente equivalentes a cada uma das seguintes.

4. $A \& B$
5. $A \rightarrow B$
6. $A \leftrightarrow B$

O *traço de Sheffer* (Sheffer stroke) é um conectivo lógico com a seguinte tabela-verdade característica:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\vert\mathcal{B}$
1	1	0
1	0	1
0	1	1
0	0	1

7. Escreva uma sentença usando os conectivos de SL que seja logicamente equivalente a $(A|B)$.

Toda sentença escrita com um conectivo de SL pode ser reescrita como uma sentença logicamente equivalente usando um ou mais traços de Sheffer. Usando apenas o traço de Sheffer, escreva sentenças equivalentes a cada uma das seguintes.

8. $\neg A$
9. $(A \& B)$
10. $(A \vee B)$
11. $(A \rightarrow B)$
12. $(A \leftrightarrow B)$

Chapter 4

Lógica quantificada

Este capítulo introduz uma linguagem lógica chamada QL. Ela é uma versão de *lógica quantificada*, porque permite quantificadores como *todos* e *alguns*. A lógica quantificada também é chamada, às vezes, de *lógica de predicados*, porque as unidades básicas da linguagem são predicados e termos.

4.1 Das sentenças aos predicados

Considere o seguinte argumento, que é obviamente válido em inglês:

> Se todo mundo conhece lógica, então ou ninguém ficará confuso ou todo mundo ficará. Todo mundo ficará confuso somente se tentarmos acreditar em uma contradição. Esta é uma aula de lógica, então todo mundo conhece lógica.
> $\therefore$ Se não tentarmos acreditar em uma contradição, então ninguém ficará confuso.

Para simbolizar esse argumento em SL, vamos precisar de uma chave de simbolização.

L: Todo mundo conhece lógica.
N: Ninguém ficará confuso.
E: Todo mundo ficará confuso.
B: Tentamos acreditar em uma contradição.

Note que N e E tratam ambos de pessoas estarem confusas, mas são duas letras sentenciais diferentes. Não podemos substituir E por $\neg N$. Por quê? $\neg N$ significa 'Não é o caso que ninguém ficará confuso.' Isso seria verdadeiro se ao menos uma pessoa ficasse confusa, o que está bem longe de dizer que *todo mundo* ficará confuso.

Uma vez que usamos letras sentenciais separadas para N e E, apagamos qualquer conexão entre as duas. Elas passam a ser apenas duas sentenças atômicas que podem ser verdadeiras ou falsas de maneira independente. Em inglês, é impossível que ao mesmo tempo ninguém e todo mundo esteja confuso. Como sentenças de SL, porém, existe uma atribuição de valores de verdade para a qual N e E são ambas verdadeiras.

Expressões como 'no one', 'everyone' e 'anyone' são chamadas de *quantifiers*. Ao traduzir N e E como sentenças atômicas separadas, deixamos de fora a *quantifier structure* (estrutura quantificacional) das sentenças. Felizmente, a estrutura quantificacional não é o que torna esse argumento válido. Portanto, podemos ignorá-la com segurança. Para ver isso, traduzimos o argumento para SL:

$$\begin{array}{rl} & L \rightarrow (N \vee E) \\ & E \rightarrow B \\ & L \\ \therefore & \neg B \rightarrow N \end{array}$$

Esse é um argumento válido em SL. (Você pode fazer uma tabela-verdade para verificar.)

Agora considere outro argumento. Este também é válido em inglês.

> Willard é um lógico. Todos os lógicos usam chapéus engraçados.
> ∴ Willard usa um chapéu engraçado.

Para simbolizá-lo em SL, definimos a seguinte chave de simbolização:

L: Willard é um lógico.
A: Todos os lógicos usam chapéus engraçados.
F: Willard usa um chapéu engraçado.

Agora simbolizamos o argumento:

$$\begin{array}{rl} & L \\ & A \\ \therefore & F \end{array}$$

Esse argumento é *inválido* em SL. (Novamente, você pode confirmar isso com uma tabela-verdade.) Há algo de muito errado aqui, porque em inglês esse é claramente um argumento válido. A simbolização em SL deixou de fora toda a estrutura importante. Mais uma vez, a tradução para SL ignora a estrutura quantificacional: a sentença 'Todos os lógicos usam chapéus engraçados' fala tanto de lógicos quanto do fato de usar chapéus engraçados. Ao não traduzir essa estrutura, perdemos a conexão entre o fato de Willard ser lógico e o fato de Willard usar chapéu.

Alguns argumentos com estrutura quantificacional podem ser capturados em SL, como o primeiro exemplo, mesmo que SL ignore essa estrutura. Outros argumentos são completamente estragados em SL, como o segundo exemplo. Note que o problema não é termos cometido algum erro ao simbolizar o segundo argumento. Essas são as melhores simbolizações que podemos dar para esses argumentos *em SL*.

De modo geral, se um argumento que contém quantificadores sai *válido em SL*, então o argumento em linguagem natural é válido. Se o argumento sai *inválido em SL*, então não podemos concluir que o argumento em linguagem natural seja inválido. Ele pode ser válido por causa de uma estrutura quantificacional que o argumento em linguagem natural tem e que o argumento em SL não tem.

De forma semelhante, se uma sentença com quantificadores sai como uma *tautologia em SL*, então a sentença em inglês é logicamente verdadeira. Se ela sai como *contingente em SL*, isso pode ser devido à estrutura quantificacional que é perdida quando traduzimos para a linguagem formal.

Para simbolizar argumentos que dependem da estrutura quantificacional, precisamos desenvolver uma linguagem lógica diferente. Chamaremos essa linguagem de lógica quantificada, QL.

4.2 Blocos básicos de QL

Assim como as sentenças eram a unidade básica da lógica sentencial, os predicados serão a unidade básica da lógica quantificada. Um predicado é uma expressão como 'é um cachorro'. Essa expressão, por si só, não é uma sentença. Ela não é nem verdadeira nem falsa. Para que seja verdadeira ou falsa, precisamos especificar algo: quem ou o que é esse ser que é um cachorro?

Os detalhes disso serão explicados ao longo do capítulo, mas aqui está a ideia básica: em QL, representaremos predicados com letras maiúsculas. Por exemplo, podemos deixar D representar '_____ é um cachorro'. Usaremos letras minúsculas como nomes de coisas específicas. Por exemplo, podemos deixar b representar Bertie. A expressão Db será uma sentença em QL. Ela é a tradução da sentença 'Bertie é um cachorro'.

Para representar a estrutura quantificacional, também teremos símbolos que representam quantificadores. Por exemplo, '$\exists$' significará 'Existe algum_____'. Assim, para dizer que existe um cachorro, podemos escrever $\exists x Dx$; isto é: existe algum x tal que x é um cachorro.

Isso virá mais adiante. Começaremos definindo termos singulares e predicados.

Termos singulares

Em inglês, um SINGULAR TERM é uma palavra ou expressão que se refere a uma pessoa, lugar ou coisa *específica*. A palavra 'dog' não é um termo singular, porque há muitos cachorros. A expressão 'Philip's dog Bertie' é um termo singular, porque se refere a um terrierzinho específico.

Um PROPER NAME (nome próprio) é um termo singular que seleciona um indivíduo sem descrevê-lo. O nome 'Emerson' é um nome próprio, e o nome, por si só, não diz nada sobre Emerson. Claro, alguns nomes são tradicionalmente dados a meninos e outros, a meninas. Se 'Jack Hathaway' é usado como termo singular, você pode supor que se refere a um homem. No entanto, o nome não significa necessariamente que a pessoa referida seja um homem ou mesmo que a criatura referida seja uma pessoa. Jack poderia ser uma girafa, e você não teria como saber isso apenas pelo nome. Há muita discussão filosófica em torno dessa questão, mas o ponto importante aqui é que um nome é um termo singular porque seleciona um único indivíduo específico.

Outros termos singulares transmitem de modo mais óbvio informações sobre aquilo a que se referem. Por exemplo, você pode saber, sem que lhe digam mais nada, que 'Philip's dog Bertie' é um termo singular que se refere a um cachorro. Uma DEFINITE DESCRIPTION (descrição definida) seleciona um indivíduo por meio de uma descrição única. Em inglês, descrições definidas costumam ser expressões do tipo 'the such-and-so'. Elas se referem a *a* coisa específica que satisfaz a descrição dada. Por exemplo, 'the tallest member of Monty Python' e 'the first emperor of China' são descrições definidas. Uma descrição que não seleciona um indivíduo específico não é uma descrição definida. 'A member of Monty Python' e 'an emperor of China' não são descrições definidas.

Podemos usar nomes próprios e descrições definidas para nos referirmos à mesma coisa. O nome próprio 'Mount Rainier' nomeia o lugar selecionado pela descrição definida 'the highest peak in Washington state'. As expressões referem-se ao mesmo lugar de maneiras diferentes. Você não aprende muita coisa quando alguém diz que está indo para Mount Rainier, a não ser que já conheça um pouco de geografia. Você até poderia supor que se trata de uma montanha, mas nem isso é garantido; por tudo o que você sabe, poderia ser uma faculdade, como Mount Holyoke. Já se eu disser que estou indo para the highest peak in Washington state, você sabe imediatamente que estou indo para uma montanha no estado de Washington.

Em inglês, a especificação de um termo singular pode depender do contexto; 'Willard' significa uma pessoa específica e não apenas alguém chamado Willard; 'P.D. Magnus', como termo singular lógico, significa *eu* e não outro P.D. Magnus. Convivemos com esse tipo de ambiguidade em inglês, mas é importante ter em mente que termos singulares em QL devem se referir a exatamente uma coisa específica.

Em QL, simbolizaremos termos singulares com letras minúsculas de a a w. Podemos adicionar subscritos se quisermos usar alguma letra mais de uma vez. Assim, $a, b, c, \ldots w, a_1, f_{32}, j_{390}$ e m_{12} são todos termos em QL.

Termos singulares são chamados de CONSTANTS porque selecionam indivíduos específicos. Note que x, y e z não são constantes em QL. Eles serão VARIABLES, letras que não representam nenhuma coisa específica. Iremos precisar delas quando introduzirmos os quantificadores.

Predicados

Os predicados mais simples são propriedades de indivíduos. Eles são coisas que você pode dizer sobre um objeto. '_____ é um cachorro' e '_____ é um integrante do Monty Python' são ambos predicados. Ao traduzir sentenças do inglês, o termo nem sempre aparece no começo da sentença: 'Um piano caiu sobre _____' também é um predicado. Predicados como esses são chamados de ONE-PLACE ou MONADIC, porque há apenas um espaço em branco a ser preenchido. Um predicado monádico e um termo singular se combinam para formar uma sentença.

Outros predicados expressam a *relação* entre duas coisas. Por exemplo, '_____ é maior do que _____', '_____ está à esquerda de _____' e '_____ deve dinheiro a _____'. Esses são predicados TWO-PLACE ou DYADIC, porque precisam ser preenchidos com dois termos para formar uma sentença.

De modo geral, você pode pensar em predicados como sentenças esquemáticas que precisam ser completadas com um certo número de termos. De maneira inversa, você pode começar com sentenças e obter predicados a partir delas removendo termos. Considere a sentença 'Vinnie pegou o carro da família emprestado de Nunzio.' Removendo um termo singular, podemos reconhecer que essa sentença usa qualquer um de três diferentes predicados monádicos:

_____pegou o carro da família emprestado de Nunzio.
Vinnie pegou _____emprestado de Nunzio.
Vinnie pegou o carro da família emprestado de _____.

Removendo dois termos singulares, podemos reconhecer três diferentes predicados diádicos:

Vinnie pegou _____emprestado de _____.
_____pegou o carro da família emprestado de _____.
_____pegou _____emprestado de Nunzio.

Removendo todos os três termos singulares, obtemos um THREE-PLACE ou TRI-

ADIC predicate:

_____pegou _____emprestado de _____.

Se estivermos traduzindo essa sentença para QL, devemos traduzi-la com um predicado de uma, duas ou três posições? Depende do que queremos ser capazes de dizer. Se a única coisa que precisarmos discutir é o fato de o carro da família ser emprestado, então a generalidade de um predicado triádico é desnecessária. Se a única coisa que precisarmos simbolizar for pessoas diferentes pegando emprestado o carro da família de Nunzio, então um predicado monádico basta.

Em geral, podemos ter predicados com quantas posições forem necessárias. Predicados com mais de uma posição são chamados POLYADIC. Predicados com n posições, para algum número n, são chamados N-PLACE ou N-ADIC.

Em QL, simbolizamos predicados com letras maiúsculas de A a Z, com ou sem subscritos. Quando fornecermos uma chave de simbolização para predicados, não usaremos espaços em branco; em vez disso, usaremos variáveis. Por convenção, constantes são listadas ao final da chave. Assim, podemos escrever uma chave como:

Ax: x está com raiva.
Hx: x está feliz.
$\mathbf{T_1xy}$: x é tão alto quanto, ou mais alto do que, y.
$\mathbf{T_2xy}$: x é tão durão quanto, ou mais durão do que, y.
Bxyz: y está entre x e z.
d: Donald
g: Gregor
m: Marybeth

Podemos simbolizar sentenças que usem qualquer combinação desses predicados e termos. Por exemplo:

1. Donald está com raiva.
2. Se Donald está com raiva, então Gregor e Marybeth também estão.
3. Marybeth é pelo menos tão alta e tão durona quanto Gregor.
4. Donald é mais baixo do que Gregor.
5. Gregor está entre Donald e Marybeth.

A sentença 1 é direta: Ad. O 'x' na entrada da chave 'Ax' é apenas um marcador de lugar; podemos substituí-lo por outros termos ao traduzir.

A sentença 2 pode ser parafraseada como 'Se Ad, então Ag e Am.' QL possui todos os conectivos verofuncionais de SL, então traduzimos isso como $Ad \rightarrow (Ag \,\&\, Am)$.

A sentença 3 pode ser traduzida como $T_1mg \,\&\, T_2mg$.

A sentença 4 pode parecer exigir um novo predicado. Se precisássemos simbolizar apenas essa sentença, poderíamos definir um predicado como Sxy para significar 'x é mais baixo do que y'. No entanto, isso ignoraria a conexão lógica entre 'mais baixo' e 'mais alto'. Considerados apenas como símbolos de QL, não há conexão entre S e T_1. Eles poderiam significar qualquer coisa. Em vez de introduzir um novo predicado, parafraseamos 4 usando predicados já presentes na chave: 'Não é o caso que Donald é tão alto quanto, ou mais alto do que, Gregor.' Podemos traduzir como $\neg T_1dg$.

A sentença 5 exige que prestemos atenção cuidadosa à ordem dos termos na chave. Ela se torna $Bdgm$.

4.3 Quantificadores

Agora estamos prontos para introduzir os quantificadores. Considere estas sentenças:

6. Todo mundo está feliz.
7. Todo mundo é pelo menos tão durão quanto Donald.
8. Alguém está com raiva.

Pode ser tentador traduzir a sentença 6 como $Hd \,\&\, Hg \,\&\, Hm$. Mas isso diria apenas que Donald, Gregor e Marybeth estão felizes. Queremos dizer que *todo mundo* está feliz, mesmo que não tenhamos definido constantes para nomear todas essas pessoas. Para isso, introduzimos o símbolo '$\forall$'. Ele é chamado de UNIVERSAL QUANTIFIER (quantificador universal).

Um quantificador deve sempre ser seguido por uma variável e por uma fórmula que contenha essa variável. Podemos traduzir a sentença 6 como $\forall xHx$. Em português, podemos parafrasear como 'Para todo x, x está feliz.' Chamamos $\forall x$ de um *x-quantifier*. A fórmula que vem depois do quantificador é chamada de *scope* (escopo) do quantificador. Daremos uma definição formal de escopo mais adiante, mas intuitivamente trata-se da parte da sentença sobre a qual o quantificador quantifica. Em $\forall xHx$, o escopo do quantificador universal é Hx.

A sentença 7 pode ser parafraseada como 'Para todo x, x é pelo menos tão durão quanto Donald.' Isso se traduz como $\forall xT_2xd$.

Nessas sentenças quantificadas, a variável x funciona como um tipo de marcador de lugar. A expressão $\forall x$ significa que você pode escolher qualquer um e colocá-lo no lugar de x. Não há motivo especial para usar x em vez de outra variável. A sentença $\forall xHx$ significa exatamente a mesma coisa que $\forall yHy$, $\forall zHz$ e $\forall x_5Hx_5$.

Para traduzir a sentença 8, introduzimos outro símbolo novo: o EXISTENTIAL QUANTIFIER (quantificador existencial), $\exists$. Assim como o quantificador universal, o quantificador existencial exige uma variável. A sentença 8 pode ser traduzida como $\exists xAx$. Isso significa que existe algum x que está com raiva. Mais precisamente, significa que existe *pelo menos uma* pessoa com raiva. Mais uma vez, a variável é um marcador de lugar; poderíamos ter traduzido igualmente como $\exists zAz$.

Considere agora estas sentenças:

9. Ninguém está com raiva.
10. Existe alguém que não está feliz.
11. Nem todo mundo está feliz.

A sentença 9 pode ser parafraseada como 'Não é o caso que alguém está com raiva.' Isso pode ser traduzido usando negação e um quantificador existencial: $\neg\exists xAx$. No entanto, a sentença 9 também pode ser parafraseada como 'Todo mundo não está com raiva.' Com isso em mente, pode ser traduzida usando negação e um quantificador universal: $\forall x\neg Ax$. Ambas são traduções aceitáveis, porque são logicamente equivalentes. O ponto crítico é se a negação vem antes ou depois do quantificador.

Em geral, $\forall x\mathcal{A}$ é logicamente equivalente a $\neg\exists x\neg\mathcal{A}$. Isso significa que qualquer sentença que possa ser simbolizada com um quantificador universal pode ser simbolizada com um quantificador existencial, e vice-versa. Uma tradução pode parecer mais natural que a outra, mas não há diferença lógica em traduzir com um quantificador ou com o outro. Para algumas sentenças, será apenas uma questão de gosto.

A sentença 10 é mais naturalmente parafraseada como 'Existe algum x tal que x não está feliz.' Isso se torna $\exists x\neg Hx$. De modo equivalente, poderíamos escrever $\neg\forall xHx$.

A sentença 11 é mais naturalmente traduzida como $\neg\forall xHx$. Ela é logicamente equivalente à sentença 10 e, portanto, também poderia ser traduzida como $\exists x\neg Hx$.

Embora tenhamos dois quantificadores em QL, poderíamos ter uma linguagem formal equivalente com apenas um quantificador. Poderíamos trabalhar apenas com o quantificador universal, por exemplo, e tratar o quantificador existencial como uma convenção notacional. Usamos colchetes [ ] para tornar algumas sentenças mais legíveis, mas sabemos que eles são, na verdade, apenas parênteses ( ). Do mesmo modo, poderíamos escrever '$\exists x$' sabendo que isso é apenas uma abreviação de '$\neg\forall x\neg$'. Há uma escolha entre tornar a lógica formalmente simples e torná-la expressivamente simples. Em QL, optamos pela simplicidade expressiva. Tanto $\forall$ quanto $\exists$ serão símbolos de QL.

Universo de discurso

Dada a chave de simbolização que estamos usando, $\forall x Hx$ significa 'Todo mundo está feliz.' Quem está incluído nesse 'todo mundo'? Quando usamos sentenças assim em português, normalmente não queremos dizer todo mundo vivo na Terra neste momento. Certamente não queremos dizer todo mundo que já viveu ou que ainda viverá. Queremos algo mais modesto: todos no prédio, todos na turma, todos na sala.

Para eliminar essa ambiguidade, precisaremos especificar um UNIVERSE OF DISCOURSE abreviado UD. O UD é o conjunto das coisas sobre as quais estamos falando. Se quisermos falar de pessoas em Chicago, definimos o UD como sendo 'pessoas em Chicago'. Escrevemos isso no início da chave de simbolização, assim:

UD: pessoas em Chicago

Os quantificadores *varrem* (range over) o universo de discurso. Dado esse UD, $\forall x$ significa 'Todo mundo em Chicago' e $\exists x$ significa 'Alguém em Chicago'. Cada constante nomeia algum membro do UD, então só podemos usar esse UD com a chave de simbolização acima se Donald, Gregor e Marybeth estiverem todos em Chicago. Se quisermos falar de pessoas em outros lugares além de Chicago, precisamos incluí-las no UD.

Em QL, o UD deve ser *non-empty*; isto é, deve conter pelo menos uma coisa. É possível construir linguagens formais que permitam UDs vazios, mas isso introduz complicações.

Mesmo permitir um UD com apenas um membro já pode produzir resultados estranhos. Suponha que tenhamos a seguinte chave de simbolização:

UD: a Torre Eiffel
Px: x está em Paris.

A sentença $\forall x Px$ poderia ser parafraseada em português como 'Tudo está em Paris.' Mas isso seria enganoso. Ela significa que tudo o que está no *UD* está em Paris. Como esse UD contém apenas a Torre Eiffel, com essa chave de simbolização $\forall x Px$ simplesmente significa que a Torre Eiffel está em Paris.

Termos não referenciais

Em QL, cada constante deve selecionar exatamente um membro do UD. Uma constante não pode se referir a mais de uma coisa ela é um termo *singular*. Ainda assim, cada constante deve selecionar *alguma* coisa. Isso está ligado a um problema filosófico clássico: o chamado problema dos *non-referring terms* (termos não referenciais).

Filósofos medievais costumavam usar sentenças sobre a *chimera* para exemplificar esse problema. Chimera é uma criatura mitológica; ela não existe de fato. Considere estas duas sentenças:

12. Chimera está com raiva.
13. Chimera não está com raiva.

É tentador simplesmente definir uma constante para significar 'chimera'. A chave de simbolização ficaria assim:

UD: criaturas na Terra

Ax: x está com raiva.

c: chimera

Poderíamos então traduzir a sentença 12 como Ac e a sentença 13 como $\neg Ac$.

Problemas surgem quando perguntamos se essas sentenças são verdadeiras ou falsas.

Uma opção seria dizer que a sentença 12 não é verdadeira, porque não existe chimera. Se a sentença 12 é falsa porque fala de algo inexistente, então a sentença 13 também deve ser falsa pela mesma razão. Mas isso significaria que Ac e $\neg Ac$ seriam ambas falsas. Dadas as condições de verdade da negação, isso não pode acontecer.

Como não podemos dizer que ambas são falsas, o que devemos fazer? Outra opção seria dizer que a sentença 12 é *meaningless* (sem sentido) por falar de uma coisa que não existe. Então Ac seria uma expressão significativa em QL para algumas interpretações, mas não para outras. Isso, porém, tornaria nossa linguagem formal dependente de interpretações particulares. Como nos interessa a forma lógica, queremos considerar a força lógica de uma sentença como Ac independentemente de qualquer interpretação específica. Se Ac fosse às vezes significativa e às vezes sem sentido, não conseguiríamos fazer isso.

Esse é o *problema dos termos não referenciais* (problem of non-referring terms), e voltaremos a ele mais adiante (veja p. 73.) O ponto importante, por enquanto, é que cada constante de QL *must* se referir a algo no UD, embora o UD possa ser qualquer conjunto de coisas que quisermos. Se quisermos simbolizar argumentos sobre criaturas mitológicas, então precisamos definir um UD que as inclua. Essa opção é importante se quisermos levar em conta a lógica de histórias. Podemos traduzir uma sentença como 'Sherlock Holmes morava no 221B Baker Street' incluindo personagens fictícios como Sherlock Holmes em nosso UD.

4.4 Traduzindo para QL

Agora já temos todas as peças de QL. Traduzir sentenças mais complicadas será apenas uma questão de saber a maneira correta de combinar predicados, constantes, quantificadores, variáveis e conectivos. Considere estas sentenças:

14. Toda moeda no meu bolso é uma *quarter* (moeda de 25 centavos de dólar).
15. Alguma moeda sobre a mesa é uma *dime* (moeda de 10 centavos de dólar).
16. Nem todas as moedas sobre a mesa são *dimes*.
17. Nenhuma das moedas no meu bolso é uma *dime*.

Ao fornecer uma chave de simbolização, precisamos especificar um UD. Como estamos falando de moedas no meu bolso e sobre a mesa, o UD deve conter pelo menos todas essas moedas. Como não estamos falando de nada além de moedas, deixamos o UD ser todas as moedas. Como não estamos falando de moedas específicas, não precisamos definir constantes. Então definimos esta chave:

UD: todas as moedas
Px: x está no meu bolso.
Tx: x está sobre a mesa.
Qx: x é uma *quarter*.
Dx: x é uma *dime*.

A sentença 14 é mais naturalmente traduzida com um quantificador universal. O quantificador universal diz algo sobre tudo no UD, não apenas sobre as moedas no meu bolso. A sentença 14 significa que (para qualquer moeda) *se* essa moeda está no meu bolso, *then* ela é uma *quarter*. Assim, podemos traduzi-la como $\forall x(Px \rightarrow Qx)$.

Como a sentença 14 fala de moedas que estão ao mesmo tempo no meu bolso *e* que são *quarters*, pode ser tentador traduzi-la usando uma conjunção. No entanto, a sentença $\forall x(Px \,\&\, Qx)$ significaria que tudo no UD está no meu bolso e é uma *quarter*: todas as moedas que existem são *quarters* no meu bolso. Isso seria algo muito estranho de se dizer, e significa algo bem diferente da sentença 14.

A sentença 15 é mais naturalmente traduzida com um quantificador existencial. Ela diz que existe alguma moeda que está sobre a mesa e que é uma *dime*. Então podemos traduzi-la como $\exists x(Tx \,\&\, Dx)$.

Note que precisamos usar um condicional com o quantificador universal, mas usamos uma conjunção com o quantificador existencial. O que significaria escrever $\exists x(Tx \rightarrow Dx)$? Provavelmente não o que você imagina. Isso significa que existe algum membro do UD que satisfaz a subfórmula; grosso modo, existe algum a tal que $(Ta \rightarrow Da)$ é verdadeiro. Em SL, $\mathcal{A} \rightarrow \mathcal{B}$ é logicamente equivalente a $\neg\mathcal{A} \vee \mathcal{B}$, e isso também valerá em QL. Assim, $\exists x(Tx \rightarrow Dx)$ é verdadeiro se existe algum a tal que $(\neg Ta \vee Da)$; isto é, é verdadeiro se alguma moeda é *ou* não está sobre a mesa *ou* é uma *dime*. É claro que existe alguma moeda que não está sobre a mesa há moedas em muitos outros lugares. Portanto, $\exists x(Tx \rightarrow Dx)$ é trivialmente verdadeiro. Um condicional normalmente será o conectivo natural a ser usado com um quantificador universal, mas um condicional no escopo de um quantificador existencial pode produzir coisas muito estranhas. Como regra geral, não coloque condicionais no escopo de quantificadores existenciais a menos que você tenha certeza de que precisa de um.

A sentença 16 pode ser parafraseada como: 'Não é o caso que toda moeda sobre a mesa é uma *dime*.' Assim, podemos traduzi-la como $\neg\forall x(Tx \rightarrow Dx)$. Você poderia olhar para a sentença 16 e parafraseá-la em vez disso como: 'Alguma moeda sobre a mesa não é uma *dime*.' Então a traduziria como $\exists x(Tx \,\&\, \neg Dx)$. Embora isso provavelmente não seja óbvio, essas duas traduções são logicamente equivalentes. (Isso decorre da equivalência lógica entre $\neg\forall x\mathcal{A}$ e $\exists x\neg\mathcal{A}$, juntamente com a equivalência entre $\neg(\mathcal{A} \rightarrow \mathcal{B})$ e $\mathcal{A} \,\&\, \neg\mathcal{B}$.)

A sentença 17 pode ser parafraseada como: 'Não é o caso que exista alguma *dime* no meu bolso.' Isso pode ser traduzido como $\neg\exists x(Px \,\&\, Dx)$. Ela também pode ser parafraseada como: 'Tudo que está no meu bolso é uma não-*dime*', e então poderia ser traduzida como $\forall x(Px \rightarrow \neg Dx)$. Mais uma vez, as duas traduções são logicamente equivalentes. Ambas são traduções corretas da sentença 17.

Agora podemos traduzir o argumento da p. 48, aquele que motivou a necessidade de quantificadores:

> Willard é um lógico. Todos os lógicos usam chapéus engraçados.
> ∴ Willard usa um chapéu engraçado.

UD: pessoas
Lx: x é um lógico.
Fx: x usa um chapéu engraçado.
w: Willard

Traduzindo, obtemos:

$$
\begin{array}{rl}
 & Lw \\
 & \forall x(Lx \rightarrow Fx) \\
\therefore & Fw
\end{array}
$$

Isto captura a estrutura que havia sido deixada de fora na tradução em SL desse argumento, e este é um argumento válido em QL.

Predicados vazios

Um predicado não precisa se aplicar a nada no UD. Um predicado que não se aplica a nada no UD é chamado de predicado VAZIO.

Suponha que queiramos simbolizar estas duas sentenças:

18. Todo macaco sabe linguagem de sinais.
19. Algum macaco sabe linguagem de sinais.

É possível escrever a chave de simbolização para essas sentenças da seguinte maneira:

UD: animais
Mx: x é um macaco.
Sx: x sabe linguagem de sinais.

A sentença 18 pode agora ser traduzida como $\forall x(Mx \rightarrow Sx)$.

A sentença 19 torna-se $\exists x(Mx \,\&\, Sx)$.

É tentador dizer que a sentença 18 implica a sentença 19; isto é: se todo macaco sabe linguagem de sinais, então deve ser o caso que algum macaco sabe linguagem de sinais. Essa é uma inferência válida na lógica aristotélica: Todos os M são S, $\therefore$ algum M é S. No entanto, a implicação não vale em QL. É possível que a sentença $\forall x(Mx \rightarrow Sx)$ seja verdadeira mesmo que a sentença $\exists x(Mx \,\&\, Sx)$ seja falsa.

Como isso pode acontecer? A resposta vem de considerar se essas sentenças seriam verdadeiras ou falsas *se não houvesse macacos*.

Definimos $\forall$ e $\exists$ de maneira que $\forall\mathcal{A}$ seja equivalente a $\neg\exists\neg\mathcal{A}$. Assim, o quantificador universal não envolve a existência de nada apenas a não existência. Se a sentença 18 é verdadeira, então *não* há macacos que não saibam linguagem de sinais. Se não houvesse macacos, então $\forall x(Mx \rightarrow Sx)$ seria verdadeira e $\exists x(Mx \,\&\, Sx)$ seria falsa.

Permitimos predicados vazios porque queremos poder dizer coisas como: 'Não sei se há algum macaco, mas quaisquer macacos que houver sabem linguagem de sinais.' Isto é, queremos poder ter predicados que não (ou possam não) se referir a nada.

O que acontece se adicionarmos um predicado vazio R à interpretação acima? Por exemplo, poderíamos definir Rx como 'x é uma geladeira'. Agora a sentença $\forall x(Rx \rightarrow Mx)$ será verdadeira. Isso é contra-intuitivo, já que não queremos dizer que há um monte de macacos-geladeira. É importante lembrar, porém, que $\forall x(Rx \rightarrow Mx)$ significa que qualquer membro do UD que seja uma geladeira é um macaco. Como o UD é animais, não há geladeiras no UD e, portanto, a sentença é trivialmente verdadeira.

Se você realmente estivesse traduzindo a sentença 'Todas as geladeiras são macacos', então precisaria incluir eletrodomésticos no UD. Nesse caso, o predicado R não seria vazio e a sentença $\forall x(Rx \rightarrow Mx)$ seria falsa.

- ▷ Um UD deve ter *pelo menos* um membro.
- ▷ Um predicado pode se aplicar a alguns, a todos ou a nenhum membro do UD.
- ▷ Uma constante deve selecionar *exatamente* um membro do UD. Um membro do UD pode ser selecionado por uma constante, por várias constantes ou por nenhuma.

Escolhendo um universo de discurso

A simbolização adequada de uma sentença em língua portuguesa em QL dependerá da chave de simbolização. De certo modo, isso é óbvio: importa se Dx significa 'x é delicado' ou 'x é perigoso'. O significado de sentenças em QL também depende do UD.

Seja Rx 'x é uma rosa', seja Tx 'x tem um espinho', e considere esta sentença:

20. Toda rosa tem um espinho.

É tentador dizer que a sentença 20 deve ser traduzida como $\forall x(Rx \rightarrow Tx)$. Se o UD contiver todas as rosas, isso estaria correto. Ainda assim, se o UD for apenas *coisas sobre a minha mesa da cozinha*, então $\forall x(Rx \rightarrow Tx)$ significará apenas que toda rosa sobre a minha mesa da cozinha tem um espinho. Se não houver rosas sobre a minha mesa, a sentença será trivialmente verdadeira.

O quantificador universal só percorre os membros do UD, então precisamos incluir todas as rosas no UD para traduzir a sentença 20. Temos duas opções. Primeiro, podemos restringir o UD para incluir todas as rosas, mas *apenas* rosas. Nesse caso, a sentença 20 se torna $\forall xTx$. Isso significa que tudo no UD tem um espinho; como o UD é justamente o conjunto das rosas, isso significa que toda rosa tem um espinho. Essa opção pode nos poupar trabalho se todas as sentenças que quisermos traduzir com essa chave forem sobre rosas.

Segundo, podemos deixar o UD conter coisas além de rosas: rododendros, ratos, rifles e o que mais houver. Então a sentença 20 deve ser $\forall x(Rx \rightarrow Tx)$.

Se quiséssemos que o quantificador universal significasse *tudo*, sem restrição, poderíamos tentar especificar um UD que contivesse tudo. Isso levaria a problemas. Será que 'tudo' inclui coisas que só foram imaginadas, como personagens fictícios? Por um lado, queremos ser capazes de simbolizar argumentos sobre Hamlet ou Sherlock Holmes. Então precisamos ter a opção de incluir personagens fictícios no UD. Por outro lado, nunca precisamos falar sobre todas as coisas que não existem. Isso talvez nem faça sentido. Há questões filosóficas aqui que não tentaremos tratar. Podemos evitar essas dificuldades especificando sempre o UD. Por exemplo, se quisermos falar de plantas, pessoas e cidades, então o UD pode ser 'seres vivos e lugares'.

Suponha que queiramos traduzir a sentença 20 e, com a mesma chave de simbolização, traduzir estas sentenças:

21. Esmerelda tem uma rosa no cabelo.
22. Todo mundo está zangado com Esmerelda.

Precisamos de um UD que inclua rosas (para que possamos simbolizar a sentença 20) e um UD que inclua pessoas (para que possamos traduzir as sentenças 21–22.) Eis uma chave adequada:

UD: pessoas e plantas
Px: x é uma pessoa.
Rx: x é uma rosa.
Tx: x tem um espinho.
Cxy: x está zangado com y.
Hxy: x tem y no cabelo.
e: Esmerelda

Como não temos um predicado que signifique '... tem uma rosa no cabelo', traduzir a sentença 21 exigirá uma paráfrase. A sentença diz que há uma rosa no cabelo de Esmerelda; isto é, há algo que é ao mesmo tempo uma rosa e está no cabelo de Esmerelda. Assim, obtemos: $\exists x(Rx \,\&\, Hex)$.

É tentador traduzir a sentença 22 como $\forall xCxe$. Infelizmente, isso significaria que todo membro do UD está zangado com Esmerelda tanto pessoas quanto plantas. Significaria, por exemplo, que a rosa no cabelo de Esmerelda está zangada com ela. Evidentemente, a sentença 22 não quer dizer isso.

'Todo mundo' significa toda pessoa, não todo membro do UD. Então podemos parafrasear a sentença 22 como: 'Toda pessoa está zangada com Esmerelda.' Sabemos como traduzir sentenças assim: $\forall x(Px \rightarrow Cxe)$

Em geral, o quantificador universal pode ser usado para significar 'todo mundo' se o UD contiver apenas pessoas. Se houver pessoas e outras coisas no UD, então 'todo mundo' deve ser tratado como 'toda pessoa'.

Traduzindo pronomes

Ao traduzir para QL, é importante entender a estrutura das sentenças que você quer traduzir. O que importa é a tradução final em QL, e às vezes você conseguirá passar de uma sentença em língua natural diretamente para uma sentença de QL. Outras vezes, ajuda parafrasear a sentença uma ou mais vezes. Cada paráfrase sucessiva deve aproximar a sentença original de algo que você consiga traduzir diretamente em QL.

Nos próximos exemplos, usaremos esta chave de simbolização:

UD: pessoas
Gx: x sabe tocar guitarra.
Rx: x é uma estrela do rock.
l: Lemmy

Agora considere estas sentenças:

23. Se Lemmy sabe tocar guitarra, então ele é uma estrela do rock.
24. Se uma pessoa sabe tocar guitarra, então ela é uma estrela do rock.

A sentença 23 e a sentença 24 têm o mesmo consequente ('. . . ele é uma estrela do rock'), mas não podem ser traduzidas da mesma maneira. Ajuda parafrasear as sentenças originais, substituindo pronomes por referências explícitas.

A sentença 23 pode ser parafraseada como: 'Se Lemmy sabe tocar guitarra, então *Lemmy* é uma estrela do rock.' Isso pode obviamente ser traduzido como $Gl \rightarrow Rl$.

A sentença 24 deve ser parafraseada de modo diferente: 'Se uma pessoa sabe tocar guitarra, então *essa pessoa* é uma estrela do rock.' Essa sentença não é sobre nenhuma pessoa em particular, então precisamos de uma variável. Traduzindo pela metade, podemos parafraseá-la como: 'Para qualquer pessoa x, se x sabe tocar guitarra, então x é uma estrela do rock.' Agora isso pode ser traduzido como $\forall x(Gx \rightarrow Rx)$. Isso é o mesmo que dizer: 'Todo mundo que sabe tocar guitarra é uma estrela do rock.'

Considere ainda estas sentenças:

25. Se alguém sabe tocar guitarra, então Lemmy sabe.
26. Se alguém sabe tocar guitarra, então essa pessoa é uma estrela do rock.

Essas duas sentenças têm o mesmo antecedente ('Se alguém sabe tocar guitarra. . .'), mas têm estruturas lógicas diferentes.

A sentença 25 pode ser parafraseada como: 'Se alguma pessoa sabe tocar guitarra, então Lemmy sabe tocar guitarra.' O antecedente e o consequente são sentenças separadas, de modo que ela pode ser simbolizada com um condicional como operador lógico principal: $\exists x Gx \to Gl$.

A sentença 26 pode ser parafraseada como: 'Para qualquer pessoa, se essa pessoa sabe tocar guitarra, então essa pessoa é uma estrela do rock.' Seria um erro simbolizá-la com um quantificador existencial, porque ela está falando sobre todos. A sentença é equivalente a 'Todos os guitarristas são estrelas do rock.' A melhor tradução é $\forall x(Gx \to Rx)$.

As palavras inglesas 'any' e 'anyone' devem tipicamente ser traduzidas usando quantificadores. Como estes dois exemplos mostram, às vezes exigem um quantificador existencial (como na sentença 25) e às vezes um quantificador universal (como na sentença 26). Se você tiver dificuldade para determinar qual é necessário, parafraseie a sentença com uma frase em inglês que use palavras diferentes de 'any' ou 'anyone'.

Quantificadores e escopo

Na sentença $\exists x Gx \to Gl$, o escopo do quantificador existencial é a expressão Gx. Faria diferença se o escopo do quantificador fosse a sentença toda? Isto é, a sentença $\exists x(Gx \to Gl)$ significa algo diferente?

Com a chave dada acima, $\exists x Gx \to Gl$ significa que, se existe algum guitarrista, então Lemmy é guitarrista. Já $\exists x(Gx \to Gl)$ significaria que existe alguma pessoa tal que, se essa pessoa fosse guitarrista, então Lemmy seria guitarrista. Lembre-se de que o condicional aqui é um condicional material; o condicional é verdadeiro se o antecedente é falso. Seja p a constante que denota o autor deste livro, alguém que certamente não é guitarrista. A sentença $Gp \to Gl$ é verdadeira porque Gp é falsa. Como alguém (no caso, p) satisfaz a sentença, então $\exists x(Gx \to Gl)$ é verdadeira. A sentença é verdadeira porque existe um não-guitarrista, independentemente da habilidade de Lemmy com a guitarra.

Algo estranho aconteceu quando mudamos o escopo do quantificador, porque o condicional em QL é um condicional material. Para manter o significado, precisaríamos mudar o quantificador: $\exists x Gx \to Gl$ significa a mesma coisa que $\forall x(Gx \to Gl)$, e $\exists x(Gx \to Gl)$ significa a mesma coisa que $\forall x Gx \to Gl$.

Essa estranheza não surge com outros conectivos nem quando a variável está no consequente do condicional. Por exemplo, $\exists x Gx \,\&\, Gl$ significa a mesma coisa que $\exists x(Gx \,\&\, Gl)$, e $Gl \to \exists x Gx$ significa a mesma coisa que $\exists x(Gl \to Gx)$.

Predicados ambíguos

Suponha que queiramos apenas traduzir esta sentença:

27. Adina é uma cirurgiã habilidosa.

Seja o UD pessoas, seja Kx 'x é uma cirurgiã habilidosa' e seja a Adina. A sentença 27 é simplesmente Ka.

Suponha, em vez disso, que queiramos traduzir este argumento:

> O hospital só contratará uma cirurgiã habilidosa. Todas as cirurgiãs são gananciosas. Billy é cirurgião, mas não é habilidoso. Logo, Billy é ganancioso, mas o hospital não o contratará.

Precisamos distinguir ser um *cirurgião habilidoso* de ser apenas um *cirurgião.* Então definimos esta chave de simbolização:

UD: pessoas
Gx: x é ganancioso.
Hx: o hospital contratará x.
Rx: x é cirurgião.
Kx: x é habilidoso.
b: Billy

Agora o argumento pode ser traduzido assim:

$$\begin{array}{ll} & \forall x\big[\neg(Rx \,\&\, Kx) \rightarrow \neg Hx\big] \\ & \forall x(Rx \rightarrow Gx) \\ & Rb \,\&\, \neg Kb \\ \therefore & Gb \,\&\, \neg Hb \end{array}$$

Em seguida, suponha que queiramos traduzir este argumento:

> Carol é uma cirurgiã habilidosa e uma tenista. Logo, Carol é uma tenista habilidosa.

Se começarmos com a chave de simbolização usada no argumento anterior, poderíamos acrescentar um predicado (seja Tx 'x é tenista') e uma constante (seja c Carol). Então o argumento se torna:

$$\begin{array}{ll} & (Rc \,\&\, Kc) \,\&\, Tc \\ \therefore & Tc \,\&\, Kc \end{array}$$

Essa tradução é um desastre! Ela transforma o que, em inglês, é um argumento péssimo em um argumento válido em QL. O problema é que há uma diferença entre ser *habilidoso como cirurgião* e ser *habilidoso como tenista.* Traduzir esse argumento corretamente exige dois predicados distintos, um para cada tipo de habilidade. Se deixarmos K_1x significar 'x é habilidoso como cirurgião' e K_2x significar 'x é habilidoso como tenista', então podemos simbolizar o argumento desta forma:

$$(Rc \,\&\, K_1c) \,\&\, Tc$$
$$\therefore\ Tc \,\&\, K_2c$$

Como o argumento em língua natural que ele traduz, este é inválido.

A moral desses exemplos é que você precisa ter cuidado ao simbolizar predicados de maneira ambígua. Problemas semelhantes podem surgir com predicados como *bom*, *ruim*, *grande* e *pequeno*. Assim como cirurgiões habilidosos e tenistas habilidosos têm habilidades diferentes, cães grandes, ratos grandes e grandes problemas são grandes de maneiras diferentes.

Basta ter um predicado que signifique 'x é um cirurgião habilidoso', em vez de dois predicados 'x é habilidoso' e 'x é cirurgião'? Às vezes, sim. Como mostra a sentença 27, às vezes não precisamos distinguir entre cirurgiões habilidosos e outros cirurgiões.

Precisamos sempre distinguir entre diferentes maneiras de ser habilidoso, bom, ruim ou grande? Não. Como o argumento sobre Billy mostra, às vezes só precisamos falar de um tipo de habilidade. Se você estiver traduzindo um argumento que é apenas sobre cães, é perfeitamente aceitável definir um predicado que signifique 'x é grande'. Se o UD incluir cães e ratos, porém, provavelmente será melhor fazer o predicado significar 'x é grande para um cão'.

Múltiplos quantificadores

Considere a seguinte chave de simbolização e as sentenças que a seguem:

UD: pessoas e cães
Dx: x é um cão.
Fxy: x é amigo de y.
Oxy: x é dono de y.
f: Fifi
g: Gerald

28. Fifi é um cão.
29. Gerald é dono de cachorro.
30. Alguém é dono de cachorro.
31. Todos os amigos de Gerald são donos de cachorro.
32. Todo dono de cachorro é amigo de algum dono de cachorro.

A sentença 28 é fácil: Df.

A sentença 29 pode ser parafraseada como: 'Existe um cão de que Gerald é dono.' Isso pode ser traduzido como $\exists x(Dx \,\&\, Ogx)$.

A sentença 30 pode ser parafraseada como: 'Existe algum y tal que y é dono de cachorro.' A subsentença 'y é dono de cachorro' é exatamente como a sentença 29, exceto pelo fato de que é sobre y em vez de ser sobre Gerald. Assim, podemos traduzir a sentença 30 como $\exists y \exists x(Dx \,\&\, Oyx)$.

A sentença 31 pode ser parafraseada como: 'Todo amigo de Gerald é dono de cachorro.' Traduzindo parte dessa sentença, obtemos $\forall x(Fxg \rightarrow$ 'x é dono de cachorro'). Mais uma vez, é importante reconhecer que 'x é dono de cachorro' é estruturalmente igual à sentença 29. Como já temos um quantificador sobre x, precisaremos de uma variável diferente para o quantificador existencial. Qualquer outra variável serve. Usando z, a sentença 31 pode ser traduzida como $\forall x\big[Fxg \rightarrow \exists z(Dz \,\&\, Oxz)\big]$.

A sentença 32 pode ser parafraseada como 'Para qualquer x que é dono de cachorro, existe um dono de cachorro que é amigo de x.' Traduzida parcialmente, isso se torna

$$\forall x\big[x \text{ é dono de cachorro} \rightarrow \exists y(y \text{ é dono de cachorro} \,\&\, Fxy)\big].$$

Completando a tradução, a sentença 32 torna-se

$$\forall x\big[\exists z(Dz \,\&\, Oxz) \rightarrow \exists y\big(\exists z(Dz \,\&\, Oyz) \,\&\, Fxy\big)\big].$$

Considere agora esta chave de simbolização e estas sentenças:

UD: pessoas
Lxy: x gosta de y.
i: Imre.
k: Karl.

33. Imre gosta de todo mundo de quem Karl gosta.
34. Existe alguém que gosta de todo mundo que gosta de todo mundo de quem essa pessoa gosta.

A sentença 33 pode ser parcialmente traduzida como $\forall x$(Karl gosta de $x \rightarrow$ Imre gosta de x). Isso se torna $\forall x(Lkx \rightarrow Lix)$.

A sentença 34 é quase um trava-língua. Há pouca esperança de escrever imediatamente a tradução completa, mas podemos proceder em pequenos passos. Uma tradução inicial e parcial poderia ser assim:

$\exists x$ todo mundo que gosta de todo mundo de quem x gosta é alguém de quem x gosta

A parte que permanece em inglês (ou português) é uma sentença universal, então traduzimos mais um pouco:

$$\exists x \forall y(y \text{ gosta de todo mundo de quem } x \text{ gosta} \rightarrow x \text{ gosta de } y).$$

O antecedente do condicional é estruturalmente igual à sentença 33, com y e x no lugar de Imre e Karl. Assim, a sentença 34 pode ser completamente traduzida da seguinte forma:

$$\exists x \forall y \big[\forall z (Lxz \to Lyz) \to Lxy\big]$$

Ao simbolizar sentenças com múltiplos quantificadores, é melhor proceder em pequenos passos. Parafraseie a sentença em língua natural de modo que a estrutura lógica fique facilmente simbolizável em QL. Depois, traduza em partes, substituindo a tarefa assustadora de traduzir uma sentença longa pela tarefa mais simples de traduzir fórmulas menores.

4.5 Sentenças de QL

Nesta seção, fornecemos uma definição formal de *fórmula bem formada* (wff) e de *sentença* de QL.

Expressões

Há seis tipos de símbolos em QL:

predicados com subscritos, quando necessário	$A, B, C, \ldots, Z$ $A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
constantes com subscritos, quando necessário	$a, b, c, \ldots, w$ $a_1, w_4, h_7, m_{32}, \ldots$
variáveis com subscritos, quando necessário	x, y, z $x_1, y_1, z_1, x_2, \ldots$
conectivos	$\neg, \&, \vee, \to, \leftrightarrow$
parênteses	(,)
quantificadores	$\forall, \exists$

Definimos uma EXPRESSÃO DE QL como qualquer sequência de símbolos de QL. Pegue quaisquer símbolos de QL e escreva-os em qualquer ordem: isso já é uma expressão.

Fórmulas bem formadas

Por definição, um TERMO DE QL é ou uma constante ou uma variável.

Uma FÓRMULA ATÔMICA DE QL é um predicado n-ário seguido de n termos.

Assim como fizemos em SL, daremos uma definição *recursiva* para uma wff de QL. Na verdade, a maior parte da definição será muito parecida com a definição de wff de SL: toda fórmula atômica é uma wff, e podemos construir novas wffs aplicando os conectivos sentenciais.

Poderíamos simplesmente acrescentar uma regra para cada um dos quantificadores e encerrar o assunto. Por exemplo: se $\mathcal{A}$ é uma wff, então $\forall x\mathcal{A}$ e $\exists x\mathcal{A}$ são wffs. No entanto, isso permitiria sentenças esquisitas como $\forall x\exists xDx$ e $\forall xDw$. O que essas fórmulas poderiam significar? Poderíamos adotar alguma interpretação para tais sentenças, mas, em vez disso, vamos escrever a definição de wff de modo que tais aberrações nem mesmo contem como bem formadas.

Para que $\forall x\mathcal{A}$ seja uma wff, $\mathcal{A}$ deve conter a variável x e não deve já conter um quantificador sobre x. $\forall xDw$ não será considerada uma wff porque 'x' não ocorre em Dw, e $\forall x\exists xDx$ não será considerada uma wff porque $\exists xDx$ já contém um quantificador sobre x.

1. Toda fórmula atômica é uma wff.
2. Se $\mathcal{A}$ é uma wff, então $\neg\mathcal{A}$ é uma wff.
3. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A}\,\&\,\mathcal{B})$ é uma wff.
4. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, $(\mathcal{A} \vee \mathcal{B})$ é uma wff.
5. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A} \rightarrow \mathcal{B})$ é uma wff.
6. Se $\mathcal{A}$ e $\mathcal{B}$ são wffs, então $(\mathcal{A} \leftrightarrow \mathcal{B})$ é uma wff.
7. Se $\mathcal{A}$ é uma wff, χ é uma variável, $\mathcal{A}$ contém pelo menos uma ocorrência de χ, e $\mathcal{A}$ não contém quantificadores sobre χ, então $\forall\chi\mathcal{A}$ é uma wff.
8. Se $\mathcal{A}$ é uma wff, χ é uma variável, $\mathcal{A}$ contém pelo menos uma ocorrência de χ, e $\mathcal{A}$ não contém quantificadores sobre χ, então $\exists\chi\mathcal{A}$ é uma wff.
9. Todas e somente as wffs de QL podem ser geradas por aplicações dessas regras.

Note que a 'χ' que aparece na definição acima não é a variável x. É uma *metavariável* que representa qualquer variável de QL. Assim, $\forall xAx$ é uma wff, mas também o são $\forall yAy$, $\forall zAz$, $\forall x_4Ax_4$ e $\forall z_9Az_9$.

Agora podemos dar uma definição formal de escopo: o ESCOPO de um quantificador é a subfórmula para a qual o quantificador é o operador lógico principal.

Sentenças

Uma sentença é algo que pode ser verdadeiro ou falso. Em SL, toda wff era uma sentença. Isso não será o caso em QL. Considere a seguinte chave de simbolização:

UD: pessoas
Lxy: x ama y
b: Boris

Considere a expressão Lzz. Ela é uma fórmula atômica: um predicado binário seguido de dois termos. Toda fórmula atômica é uma wff, então Lzz é uma wff. Ela significa alguma coisa? Você pode pensar que significa que z ama a si mesmo, do mesmo modo que Lbb significa que Boris ama a si mesmo. No entanto, z é uma variável; ela não nomeia uma pessoa da maneira como uma constante nomeia. A wff Lzz não nos diz como interpretar z. Ele significa todo mundo? alguém qualquer? alguma pessoa? Se tivéssemos um quantificador sobre z, isso nos diria como interpretar z. Por exemplo, $\exists zLzz$ significaria que alguém ama a si mesmo.

Algumas linguagens formais tratam uma wff como Lzz como se tivesse, implicitamente, um quantificador universal na frente. Não faremos isso em QL. Se você quer dizer que todo mundo ama a si mesmo, então precisa escrever o quantificador: $\forall zLzz$

Para dar sentido a uma variável, precisamos de um quantificador que nos diga como interpretar essa variável. O escopo de um quantificador sobre x, por exemplo, é a parte da fórmula onde o quantificador diz como interpretar x.

Para sermos precisos quanto a isso, definimos uma VARIÁVEL LIGADA como uma ocorrência de uma variável χ que está no escopo de um quantificador sobre χ. Uma VARIÁVEL LIVRE é uma ocorrência de variável que não está ligada.

Por exemplo, considere a wff $\forall x(Ex \vee Dy) \rightarrow \exists z(Ex \rightarrow Lzx)$. O escopo do quantificador universal $\forall x$ é $(Ex \vee Dy)$, de modo que a primeira ocorrência de x está ligada pelo quantificador universal, mas a segunda e a terceira ocorrência de x são livres. Não há quantificador sobre y, então o y é livre. O escopo do quantificador existencial $\exists z$ é $(Ex \rightarrow Lzx)$, de modo que ambas as ocorrências de z estão ligadas por ele.

Definimos uma SENTENÇA de QL como uma wff de QL que não contém variáveis livres.

Convenções notacionais

Adotaremos as mesmas convenções notacionais que usamos para SL (p. 31.) Em primeiro lugar, podemos omitir os parênteses mais externos de uma fórmula. Em segundo lugar, usaremos colchetes '[' e ']' no lugar de parênteses para aumentar a legibilidade das fórmulas. Em terceiro lugar, omitiremos parênteses entre cada par de conjunções ao escrever séries longas de conjunções. Em quarto lugar, omitiremos parênteses entre cada par de disjunções ao escrever séries longas de disjunções.

4.6 Identidade

Considere esta sentença:

35. Pavel deve dinheiro a todas as outras pessoas.

Seja o UD o conjunto das pessoas; isso nos permitirá traduzir ‘todo mundo’ com um quantificador universal. Seja Oxy ‘x deve dinheiro a y’, e seja p Pavel. Agora podemos simbolizar a sentença 35 como $\forall xOpx$. Infelizmente, essa tradução tem consequências estranhas. Ela diz que Pavel deve dinheiro a todo membro do UD, incluindo o próprio Pavel; em particular, implica que Pavel deve dinheiro a si mesmo. No entanto, a sentença 35 não diz que Pavel deve dinheiro a si mesmo; ele deve dinheiro a todos os *outros*. Este é um problema, porque $\forall xOpx$ é a melhor tradução que conseguimos dar para essa sentença em QL.

A solução é acrescentar outro símbolo a QL. O símbolo ‘$=$’ é um predicado binário. Como ele tem um significado lógico especial, escrevemo-lo de modo um pouco diferente: para dois termos t_1 e t_2, $t_1 = t_2$ é uma fórmula atômica.

O predicado $x = y$ significa ‘x é idêntico a y’. Isso não quer dizer apenas que x e y são indistinguíveis ou que os mesmos predicados são verdadeiros de ambos. Quer dizer que x e y são exatamente a mesma coisa.

Quando escrevemos $x \neq y$, queremos dizer que x e y não são idênticos. Não há necessidade de introduzir isso como um predicado adicional. Em vez disso, $x \neq y$ é uma abreviação de $\neg(x = y)$.

Agora suponha que queiramos simbolizar esta sentença:

36. Pavel é o senhor Checkov.

Seja a constante c o senhor Checkov. A sentença 36 pode ser simbolizada como $p = c$. Isso significa que as constantes p e c referem-se ao mesmo sujeito.

Tudo isso é muito bom, mas como isso ajuda com a sentença 35? Essa sentença pode ser parafraseada como: ‘Toda pessoa que não é Pavel é alguém a quem Pavel deve dinheiro.’ Essa é uma estrutura de sentença que já sabemos simbolizar: ‘Para todo x, se x não é Pavel, então x é alguém a quem Pavel deve dinheiro.’ Em QL com identidade, isso se torna $\forall x(x \neq p \rightarrow Opx)$.

Além de sentenças que usam a palavra ‘else’ (ou, em português, ‘outro’, ‘demais’), a identidade será útil ao simbolizar algumas sentenças que contêm as palavras ‘além de’ e ‘apenas’. Considere estes exemplos:

37. Ninguém além de Pavel deve dinheiro a Hikaru.
38. Só Pavel deve dinheiro a Hikaru.

Acrescentamos a constante h, que significa Hikaru.

A sentença 37 pode ser parafraseada como: 'Ninguém que não seja Pavel deve dinheiro a Hikaru.' Isso pode ser traduzido como $\neg\exists x(x \neq p \,\&\, Oxh)$.

A sentença 38 pode ser parafraseada como: 'Pavel deve dinheiro a Hikaru *e* ninguém além de Pavel deve dinheiro a Hikaru.' Já traduzimos um dos conjunctos, e o outro é direto. A sentença 38 torna-se $Oph \,\&\, \neg\exists x(x \neq p \,\&\, Oxh)$.

Expressões de quantidade

Também podemos usar a identidade para dizer quantas coisas há de um determinado tipo. Por exemplo, considere estas sentenças:

39. Há pelo menos uma maçã sobre a mesa.
40. Há pelo menos duas maçãs sobre a mesa.
41. Há pelo menos três maçãs sobre a mesa.

Seja o UD o conjunto das *coisas sobre a mesa*, e seja Ax 'x é uma maçã'.

A sentença 39 não exige identidade. Ela pode ser traduzida adequadamente como $\exists x Ax$: existe alguma maçã sobre a mesa talvez muitas, mas pelo menos uma.

Pode ser tentador também traduzir a sentença 40 sem identidade. No entanto, considere a sentença $\exists x\exists y(Ax \,\&\, Ay)$. Ela significa que existe alguma maçã x no UD e alguma maçã y no UD. Como nada impede que x e y escolham o mesmo membro do UD, isso seria verdadeiro mesmo que houvesse apenas uma maçã. Para garantir que haja duas maçãs *diferentes*, precisamos do predicado de identidade. A sentença 40 precisa dizer que as duas maçãs existentes não são idênticas, então pode ser traduzida como $\exists x\exists y(Ax \,\&\, Ay \,\&\, x \neq y)$.

A sentença 41 exige falar de três maçãs diferentes. Ela pode ser traduzida como $\exists x\exists y\exists z(Ax \,\&\, Ay \,\&\, Az \,\&\, x \neq y \,\&\, y \neq z \,\&\, x \neq z)$.

Continuando dessa forma, poderíamos traduzir 'Há pelo menos n maçãs sobre a mesa.' Há um resumo de como simbolizar sentenças desse tipo na p. 153.

Agora considere estas sentenças:

42. Há no máximo uma maçã sobre a mesa.
43. Há no máximo duas maçãs sobre a mesa.

A sentença 42 pode ser parafraseada como: 'Não é o caso que haja pelo menos *duas* maçãs sobre a mesa.' Isso é simplesmente a negação da sentença 40:

$$\neg\exists x\exists y(Ax \,\&\, Ay \,\&\, x \neq y)$$

A sentença 42 também pode ser abordada de outra maneira. Ela significa que quaisquer maçãs que houver sobre a mesa devem ser uma única e mesma maçã, de modo que pode ser traduzida como $\forall x \forall y\big[(Ax \,\&\, Ay) \to x = y\big]$. As duas traduções são logicamente equivalentes, portanto ambas estão corretas.

De modo semelhante, a sentença 43 pode ser traduzida de duas formas equivalentes. Ela pode ser parafraseada como: 'Não é o caso que haja *três* ou mais maçãs distintas', de modo que pode ser traduzida como a negação da sentença 41. Usando quantificadores universais, também pode ser traduzida como

$$\forall x \forall y \forall z\big[(Ax \,\&\, Ay \,\&\, Az) \to (x = y \vee x = z \vee y = z)\big].$$

Veja p. 153 para o caso geral.

Os exemplos acima são sentenças sobre maçãs, mas a estrutura lógica dessas sentenças traduz desigualdades matemáticas como $a \geq 3$, $a \leq 2$ e assim por diante. Também queremos poder traduzir afirmações de igualdade que dizem exatamente quantas coisas há. Por exemplo:

44. Há exatamente uma maçã sobre a mesa.
45. Há exatamente duas maçãs sobre a mesa.

A sentença 44 pode ser parafraseada como: 'Há *pelo menos* uma maçã sobre a mesa, e há *no máximo* uma maçã sobre a mesa.' Isso é apenas a conjunção da sentença 39 com a sentença 42: $\exists x Ax \,\&\, \forall x \forall y\big[(Ax \,\&\, Ay) \to x = y\big]$. Esse é um modo um pouco complicado de proceder. Talvez seja mais direto parafrasear a sentença 44 como: 'Há uma coisa que é a única maçã sobre a mesa.' Entendida assim, a sentença pode ser traduzida como $\exists x\big[Ax \,\&\, \neg\exists y(Ay \,\&\, x \neq y)\big]$.

De modo análogo, a sentença 45 pode ser parafraseada como: 'Há duas maçãs diferentes sobre a mesa, e essas são as únicas maçãs sobre a mesa.' Isso pode ser traduzido como $\exists x \exists y\big[Ax \,\&\, Ay \,\&\, x \neq y \,\&\, \neg\exists z(Az \,\&\, x \neq z \,\&\, y \neq z)\big]$.

Finalmente, considere esta sentença:

46. Há no máximo duas coisas sobre a mesa.

Pode ser tentador acrescentar um predicado tal que Tx signifique 'x é uma coisa sobre a mesa'. No entanto, isso é desnecessário. Como o UD é o conjunto das coisas sobre a mesa, todos os membros do UD estão sobre a mesa. Se quisermos falar de uma *coisa sobre a mesa*, basta usar um quantificador. A sentença 46 pode ser simbolizada como a sentença 43 (que dizia que havia no máximo duas maçãs), mas omitindo completamente o predicado. Isto é, a sentença 46 pode ser traduzida como $\forall x \forall y \forall z(x = y \vee x = z \vee y = z)$.

Técnicas para simbolizar expressões de quantidade ('no máximo', 'pelo menos' e 'exatamente') são resumidas na p. 153.

Descrições definidas

Lembre que uma constante de QL deve referir-se a algum membro do UD. Essa restrição nos permite evitar o problema de termos que não se referem a nada. Dados um UD que incluísse apenas criaturas realmente existentes e uma constante c que significasse 'quimera' (uma criatura mítica), sentenças contendo c se tornariam impossíveis de avaliar.

A solução mais influente para esse problema foi introduzida por Bertrand Russell em 1905. Russell perguntou como deveríamos entender a seguinte sentença:

47. O atual rei da França é careca.

A expressão 'o atual rei da França' deveria selecionar um indivíduo por meio de uma descrição definida. No entanto, não havia rei da França em 1905 e não há nenhum agora. Como a descrição é um termo que não se refere a nada, não podemos simplesmente definir uma constante para significar 'o atual rei da França' e traduzir a sentença como Kf.

A ideia de Russell era que sentenças que contêm descrições definidas têm uma estrutura lógica diferente da estrutura de sentenças que contêm nomes próprios, ainda que compartilhem a mesma forma gramatical. O que queremos dizer quando usamos uma descrição referencial sem problema, como 'o pico mais alto do estado de Washington'? Queremos dizer que há tal pico, porque não poderíamos falar dele de outro modo. Também queremos dizer que ele é o único pico desse tipo. Se houvesse outro pico no estado de Washington com altura exatamente igual à do Monte Rainier, então o Monte Rainier não seria *o* pico mais alto.

De acordo com essa análise, a sentença 47 está dizendo três coisas. Primeiro, ela faz uma afirmação de *existência*: existe um atual rei da França. Segundo, faz uma afirmação de *unicidade*: esse sujeito é o único atual rei da França. Terceiro, faz uma afirmação de *predicação*: esse sujeito é careca.

Para simbolizar descrições definidas desse modo, precisamos do predicado de identidade. Sem ele, não poderíamos traduzir a afirmação de unicidade que (segundo Russell) está implícita na descrição definida.

Seja o UD o conjunto das *pessoas atualmente vivas*, seja Fx 'x é o atual rei da França', e seja Bx 'x é careca'. A sentença 47 pode então ser traduzida como $\exists x\big[Fx \,\&\, \neg\exists y(Fy \,\&\, x \neq y) \,\&\, Bx\big]$. Isso diz que existe algum sujeito que é o atual rei da França, ele é o único atual rei da França e ele é careca.

Entendida assim, a sentença 47 é significativa, mas falsa. Ela diz que esse sujeito existe, mas ele não existe.

O problema dos termos sem referência é mais incômodo quando tentamos traduzir negações. Portanto, considere esta sentença:

48. O atual rei da França não é careca.

Segundo Russell, essa sentença é ambígua em inglês. Ela poderia significar duas coisas diferentes:

48a. Não é o caso que o atual rei da França seja careca.
48b. O atual rei da França é não-careca.

Ambos os possíveis significados negam a sentença 47, mas colocam a negação em lugares diferentes.

A sentença 48a é chamada de NEGAÇÃO DE ESCOPO AMPLO, porque nega a sentença inteira. Ela pode ser traduzida como $\neg\exists x\big[Fx \,\&\, \neg\exists y(Fy \,\&\, x \neq y) \,\&\, Bx\big]$. Isso não diz nada sobre o atual rei da França, mas afirma que certa sentença sobre o atual rei da França é falsa. Como a sentença 47 é falsa, a sentença 48a é verdadeira.

A sentença 48b diz algo sobre o atual rei da França. Ela afirma que ele não tem a propriedade de ser careca. Como a sentença 47, ela faz uma afirmação de existência e uma de unicidade; apenas nega a afirmação de predicação. Isso é chamado de NEGAÇÃO DE ESCOPO ESTREITO. Pode ser traduzida como $\exists x\big[Fx \,\&\, \neg\exists y(Fy \,\&\, x \neq y) \,\&\, \neg Bx\big]$. Como não há atual rei da França, essa sentença é falsa.

A teoria de descrições definidas de Russell resolve o problema dos termos sem referência e também explica por que ele parecia tão paradoxal. Antes de distinguirmos entre a negação de escopo amplo e a de escopo estreito, parecia que sentenças como 48 deveriam ser ao mesmo tempo verdadeiras e falsas. Ao mostrar que tais sentenças são ambíguas, Russell mostrou que elas são verdadeiras se entendidas de um modo, mas falsas se entendidas de outro.

Para uma discussão mais detalhada da teoria das descrições definidas de Russell, incluindo objeções a ela, veja a entrada de Peter Ludlow 'descriptions' em *The Stanford Encyclopedia of Philosophy*: edição do verão de 2005, editada por Edward N. Zalta, `http://plato.stanford.edu/archives/sum2005/entries/descriptions/`

Practice Exercises

⋆ **Part A** Usando a chave de simbolização dada, traduza cada sentença em inglês para QL.

UD: todos os animais
Ax: x é um jacaré.
Mx: x é um macaco.

Rx: x é um réptil.
Zx: x vive no zoológico.
Lxy: x ama y.
a: Amos
b: Bouncer
c: Cleo

1. Amos, Bouncer e Cleo vivem todos no zoológico.
2. Bouncer é um réptil, mas não é um jacaré.
3. Se Cleo ama Bouncer, então Bouncer é um macaco.
4. Se tanto Bouncer quanto Cleo são jacarés, então Amos ama os dois.
5. Algum réptil vive no zoológico.
6. Todo jacaré é um réptil.
7. Todo animal que vive no zoológico é ou um macaco ou um jacaré.
8. Há répteis que não são jacarés.
9. Cleo ama um réptil.
10. Bouncer ama todos os macacos que vivem no zoológico.
11. Todos os macacos que Amos ama o amam de volta.
12. Se algum animal é réptil, então Amos é réptil.
13. Se algum animal é jacaré, então ele é réptil.
14. Todo macaco que Cleo ama também é amado por Amos.
15. Existe um macaco que ama Bouncer, mas infelizmente Bouncer não retribui esse amor.

Part B Estas são figuras silogísticas identificadas por Aristóteles e seus sucessores, juntamente com seus nomes medievais. Traduza cada argumento para QL.

Barbara Todos os B são C. Todos os A são B. $\therefore$ Todos os A são C.

Baroco Todos os C são B. Algum A não é B. $\therefore$ Algum A não é C.

Bocardo Algum B não é C. Todos os A são B. $\therefore$ Algum A não é C.

Celantes Nenhum B é C. Todos os A são B. $\therefore$ Nenhum C é A.

Celarent Nenhum B é C. Todos os A são B. $\therefore$ Nenhum A é C.

Cemestres Nenhum C é B. Nenhum A é B. $\therefore$ Nenhum A é C.

Cesare Nenhum C é B. Todos os A são B. $\therefore$ Nenhum A é C.

Dabitis Todos os B são C. Algum A é B. $\therefore$ Algum C é A.

Darii Todos os B são C. Algum A é B. $\therefore$ Algum A é C.

Datisi Todos os B são C. Todo A é C. $\therefore$ Algum A é C.

Disamis Algum A é B. Todos os A são C. $\therefore$ Algum B é C.

Ferison Nenhum B é C. Algum A é B. $\therefore$ Algum A não é C.

Ferio Nenhum B é C. Algum A é B. $\therefore$ Algum A não é C.

Festino Nenhum C é B. Algum A é B. $\therefore$ Algum A não é C.

Baralipton Todos os B são C. Todos os A são B. $\therefore$ Algum C é A.

Frisesomorum Algum B é C. Nenhum A é B. $\therefore$ Algum C não é A.

Part C Usando a chave de simbolização dada, traduza cada sentença em inglês para QL.

UD: todos os animais
Dx: x é um cachorro.
Sx: x gosta de filmes de samurai.
Lxy: x é maior que y.
b: Bertie
e: Emerson
f: Fergis

1. Bertie é um cachorro que gosta de filmes de samurai.
2. Bertie, Emerson e Fergis são todos cachorros.
3. Emerson é maior que Bertie, e Fergis é maior que Emerson.
4. Todos os cachorros gostam de filmes de samurai.
5. Somente cachorros gostam de filmes de samurai.
6. Existe um cachorro que é maior que Emerson.
7. Se existe um cachorro maior que Fergis, então existe um cachorro maior que Emerson.
8. Nenhum animal que gosta de filmes de samurai é maior que Emerson.
9. Nenhum cachorro é maior que Fergis.
10. Qualquer animal que não gosta de filmes de samurai é maior que Bertie.
11. Existe um animal que está, em tamanho, entre Bertie e Emerson.
12. Não há cachorro que esteja, em tamanho, entre Bertie e Emerson.
13. Nenhum cachorro é maior do que ele mesmo.
14. Para todo cachorro, existe algum cachorro maior do que ele.
15. Existe um animal que é menor do que todo cachorro.
16. Se existe um animal que é maior do que qualquer cachorro, então esse animal não gosta de filmes de samurai.

Part D Para cada argumento, escreva uma chave de simbolização e traduza o argumento para QL.

1. Nada sobre a minha mesa escapa à minha atenção. Há um computador sobre a minha mesa. Logo, há um computador que não escapa à minha atenção.
2. Todos os meus sonhos são em preto e branco. Os programas antigos de TV são em preto e branco. Portanto, alguns dos meus sonhos são programas antigos de TV.

3. Nem Holmes nem Watson esteve na Austrália. Uma pessoa só poderia ver um canguru se tivesse estado na Austrália ou em um zoológico. Embora Watson não tenha visto um canguru, Holmes viu. Portanto, Holmes esteve em um zoológico.
4. Ninguém espera a Inquisição Espanhola. Ninguém conhece as dificuldades que eu tenho enfrentado. Portanto, qualquer um que espera a Inquisição Espanhola conhece as dificuldades que eu tenho enfrentado.
5. Uma antílope é maior do que uma caixa de pão. Estou pensando em algo que não é maior do que uma caixa de pão, e que é ou uma antílope ou um melão. Assim, estou pensando em um melão.
6. Todos os bebês são ilógicos. Ninguém que seja ilógico consegue controlar um crocodilo. Berthold é um bebê. Portanto, Berthold é incapaz de controlar um crocodilo.

⋆ **Part E** Usando a chave de simbolização dada, traduza cada sentença em inglês para QL.

UD: doces
Cx: x tem chocolate.
Mx: x tem marzipã.
Sx: x tem açúcar.
Tx: Boris provou x.
Bxy: x é melhor do que y.

1. Boris nunca provou nenhum doce.
2. Marzipã é sempre feito com açúcar.
3. Algum doce é sem açúcar.
4. O melhor de todos os doces é o chocolate.
5. Nenhum doce é melhor do que ele mesmo.
6. Boris nunca provou chocolate sem açúcar.
7. Boris já provou marzipã e chocolate, mas nunca juntos.
8. Qualquer doce com chocolate é melhor do que qualquer doce sem chocolate.
9. Qualquer doce com chocolate e marzipã é melhor do que qualquer doce que não tenha nem chocolate nem marzipã.

Part F Usando a chave de simbolização dada, traduza cada sentença em inglês para QL.

UD: pessoas e pratos em um potluck
Rx: x acabou.
Tx: x está na mesa.
Fx: x é comida.
Px: x é uma pessoa.
Lxy: x gosta de y.
e: Eli
f: Francesca

g: o guacamole

1. Toda a comida está sobre a mesa.
2. Se o guacamole não acabou, então ele está na mesa.
3. Todo mundo gosta do guacamole.
4. Se alguém gosta do guacamole, então Eli gosta.
5. Francesca só gosta dos pratos que já acabaram.
6. Francesca não gosta de ninguém, e ninguém gosta de Francesca.
7. Eli gosta de qualquer pessoa que goste do guacamole.
8. Eli gosta de qualquer pessoa que goste das pessoas de quem ele gosta.
9. Se já houver uma pessoa em cima da mesa, então toda a comida já deve ter acabado.

⋆ **Part G** Usando a chave de simbolização dada, traduza cada sentença em inglês para QL.

UD: pessoas
Dx: x dança balé.
Fx: x é mulher.
Mx: x é homem.
Cxy: x é filho de y.
Sxy: x é irmão de y.
e: Elmer
j: Jane
p: Patrick

1. Todos os filhos de Patrick dançam balé.
2. Jane é filha de Patrick.
3. Patrick tem uma filha.
4. Jane é filha única.
5. Todas as filhas de Patrick dançam balé.
6. Patrick não tem filhos homens.
7. Jane é sobrinha de Elmer.
8. Patrick é irmão de Elmer.
9. Os irmãos de Patrick não têm filhos.
10. Jane é tia.
11. Todo mundo que dança balé tem uma irmã que também dança balé.
12. Todo homem que dança balé é filho de alguém que dança balé.

Part H Identifique quais variáveis estão ligadas e quais estão livres.

1. $\exists x Lxy \,\&\, \forall y Lyx$
2. $\forall x Ax \,\&\, Bx$
3. $\forall x(Ax \,\&\, Bx) \,\&\, \forall y(Cx \,\&\, Dy)$
4. $\forall x \exists y[Rxy \rightarrow (Jz \,\&\, Kx)] \vee Ryx$
5. $\forall x_1(Mx_2 \leftrightarrow Lx_2x_1) \,\&\, \exists x_2 Lx_3x_2$

Part I Usando a chave de simbolização dada, traduza cada sentença em inglês para QL com identidade. A última sentença é ambígua e pode ser traduzida de duas maneiras; você deve fornecer ambas as traduções. (Dica: a identidade só é necessária para as quatro últimas sentenças.)

UD: pessoas
Kx: x conhece a combinação do cofre.
Sx: x é espião.
Vx: x é vegetariano.
Txy: x confia em y.
h: Hofthor
i: Ingmar

1. Hofthor é espião, mas nenhum vegetariano é espião.
2. Ninguém conhece a combinação do cofre, a menos que Ingmar conheça.
3. Nenhum espião conhece a combinação do cofre.
4. Nem Hofthor nem Ingmar é vegetariano.
5. Hofthor confia em um vegetariano.
6. Todo mundo que confia em Ingmar confia em um vegetariano.
7. Todo mundo que confia em Ingmar confia em alguém que confia em um vegetariano.
8. Só Ingmar conhece a combinação do cofre.
9. Ingmar confia em Hofthor, mas em mais ninguém.
10. A pessoa que conhece a combinação do cofre é vegetariana.
11. A pessoa que conhece a combinação do cofre não é espiã.

$\star$ **Part J** Usando a chave de simbolização dada, traduza cada sentença em inglês para QL com identidade. As duas últimas sentenças são ambíguas e podem ser traduzidas de duas maneiras; você deve fornecer ambas as traduções para cada uma delas.

UD: cartas de um baralho padrão
Bx: x é preta.
Cx: x é de paus.
Dx: x é um dois.
Jx: x é um valete.
Mx: x é o homem com o machado.
Ox: x tem um olho só.
Wx: x é coringa.

1. Todas as cartas de paus são pretas.
2. Não há cartas coringa.
3. Há pelo menos duas cartas de paus.
4. Há mais de um valete de um olho só.
5. Há no máximo dois valetes de um olho só.
6. Há dois valetes pretos.
7. Há quatro cartas de dois.

8. O dois de paus é uma carta preta.
9. Os valetes de um olho só e o homem com o machado são curingas.
10. Se o dois de paus for coringa, então há exatamente uma carta coringa.
11. O homem com o machado não é um valete.
12. O dois de paus não é o homem com o machado.

Part K Usando a chave de simbolização dada, traduza cada sentença em inglês para QL com identidade. As duas últimas sentenças são ambíguas e podem ser traduzidas de duas maneiras; você deve fornecer ambas as traduções para cada uma delas.

UD: animais no mundo
Bx: x está no campo do fazendeiro Brown.
Hx: x é um cavalo.
Px: x é um Pégaso.
Wx: x tem asas.

1. Há pelo menos três cavalos no mundo.
2. Há pelo menos três animais no mundo.
3. Há mais de um cavalo no campo do fazendeiro Brown.
4. Há três cavalos no campo do fazendeiro Brown.
5. Há uma única criatura alada no campo do fazendeiro Brown; quaisquer outras criaturas no campo não têm asas.
6. O Pégaso é um cavalo alado.
7. O animal no campo do fazendeiro Brown não é um cavalo.
8. O cavalo no campo do fazendeiro Brown não tem asas.

Chapter 5

Semântica formal

Neste capítulo, descrevemos uma *semântica formal* para LV e para LQ. A palavra 'semântica' vem da palavra grega para 'marca' e significa 'relacionada ao significado'. Portanto, uma semântica formal será uma conta matemática do significado na linguagem formal.

Uma linguagem formal e lógica é construída a partir de dois tipos de elementos: símbolos lógicos e símbolos não lógicos. Conectivos (como '$\&$') e quantificadores (como '$\forall$') são símbolos lógicos, porque seu significado é especificado dentro da linguagem formal. Ao escrever uma chave de simbolização, você não tem permissão para alterar o significado dos símbolos lógicos. Você não pode dizer, por exemplo, que o símbolo '$\neg$' significará 'não' em um argumento e 'talvez' em outro. O símbolo '$\neg$' sempre significa negação lógica. É usado para traduzir a palavra em português 'não', mas é um símbolo de uma linguagem formal e é definido por suas condições de verdade.

As letras de sentença em LV são símbolos não lógicos, porque seu significado não é definido pela estrutura lógica de LV. Quando traduzimos um argumento do português para LV, por exemplo, a letra de sentença M não tem seu significado fixado antecipadamente; em vez disso, fornecemos uma chave de simbolização que diz como M deve ser interpretada nesse argumento. Em LQ, os predicados e constantes são símbolos não lógicos.

Ao traduzir do português para uma linguagem formal, fornecemos chaves de simbolização que eram interpretações de todos os símbolos não lógicos que usamos na tradução. Uma INTERPRETAÇÃO dá um significado a todos os elementos não lógicos da linguagem.

É possível fornecer interpretações diferentes que não fazem diferença formal. Em LV, por exemplo, podemos dizer que D significa 'Hoje é terça-feira'; podemos dizer, em vez disso, que D significa 'Hoje é o dia depois de segunda-feira'. Essas são duas interpretações diferentes, porque usam sentenças em português diferentes para o significado de D. No entanto, formalmente, não há diferença entre elas. Tudo o que importa, uma vez que simbolizamos essas sentenças, é se elas são verdadeiras ou falsas. Para caracterizar o que faz uma diferença na linguagem formal, precisamos saber o que torna as sentenças verdadeiras ou falsas. Para isso, precisamos de uma caracterização formal de *verdade*.

Quando demos definições para uma sentença de LV e para uma sentença de LQ, distinguimos entre a LINGUAGEM OBJETO e a METALINGUAGEM. A linguagem objeto é a linguagem sobre a qual estamos *falando*: LV ou LQ. A metalinguagem é a linguagem que usamos para falar sobre a linguagem objeto: português, complementado com algum jargão matemático. Será importante manter essa distinção em mente.

5.1 Semântica para LV

Esta seção fornece uma caracterização rigorosa e formal de *verdade em LV* que se baseia no que já sabemos ao fazer tabelas-verdade. Fomos capazes de usar tabelas-verdade para testar de forma confiável se uma sentença era uma tautologia em LV, se duas sentenças eram equivalentes, se um argumento era válido e assim por diante. Por exemplo: $\mathcal{A}$ é uma tautologia em LV se for V em cada linha de uma tabela-verdade completa.

Isso funcionou porque cada linha de uma tabela-verdade corresponde a uma maneira como o mundo poderia ser. Consideramos todas as combinações possíveis de 1 e 0 para as letras de sentença que fizeram diferença para as sentenças com as quais nos importávamos. A tabela-verdade nos permitiu determinar o que aconteceria dadas essas combinações diferentes.

Uma vez que construímos uma tabela-verdade, os símbolos '1' e '0' são divorciados de seu significado metalinguístico de 'verdadeiro' e 'falso'. Interpretamos '1' como significando 'verdadeiro', mas as propriedades formais de 1 são definidas pelas tabelas-verdade características para os vários conectivos. Os símbolos em uma tabela-verdade têm um significado formal que podemos especificar inteiramente em termos de como os conectivos operam. Por exemplo, se A é valor 1, então $\neg A$ é valor 0.

Em resumo: Verdade em LV é apenas a atribuição de um 1 ou um 0.

Para definir formalmente a verdade em LV, então, queremos uma função que atribua um 1 ou 0 a cada uma das sentenças de LV. Podemos interpretar essa função como uma definição de verdade para LV se ela atribuir 1 a todas as sentenças verdadeiras de LV e 0 a todas as sentenças falsas de LV. Chame essa função de 'v' (para 'valoração'). Queremos que v seja uma função tal que para qualquer sentença $\mathcal{A}$, $v(\mathcal{A}) = 1$ se $\mathcal{A}$ for verdadeira e $v(\mathcal{A}) = 0$ se $\mathcal{A}$ for falsa.

Lembre-se de que a definição recursiva de uma fbf para LV teve dois estágios: O primeiro passo disse que sentenças atômicas (letras de sentença solitárias) são fbfs. O segundo estágio permitiu que fbfs fossem construídas a partir de fbfs mais básicas. Havia cláusulas da definição para todos os conectivos sentenciais. Por exemplo, se $\mathcal{A}$ é uma fbf, então $\neg\mathcal{A}$ é uma fbf.

Nossa estratégia para definir a função verdade, v, também será em dois passos. O primeiro passo lidará com a verdade para sentenças atômicas; o segundo passo lidará com a verdade para sentenças compostas.

Verdade em LV

Como podemos definir verdade para uma sentença atômica de LV? Considere, por exemplo, a sentença M. Sem uma interpretação, não podemos dizer se M é verdadeira ou falsa. Pode significar qualquer coisa. Se usarmos M para simbolizar 'A lua orbita a Terra', então M é verdadeira. Se usarmos M para simbolizar 'A lua é um nabo gigante', então M é falsa.

Além disso, a maneira como você descobriria se M é verdadeira ou não depende do que M significa. Se M significa 'É segunda-feira', então você precisaria verificar um calendário. Se M significa 'A lua Io de Júpiter tem atividade vulcânica significativa', então você precisaria verificar um texto de astronomia - e os astrônomos sabem porque enviaram satélites para observar Io.

Quando damos uma chave de simbolização para LV, fornecemos uma interpretação das letras de sentença que usamos. A chave dá uma sentença em português para cada letra de sentença que usamos. Dessa forma, a interpretação especifica o que cada uma das letras de sentença *significa*. No entanto, isso não é suficiente para determinar se essa sentença é verdadeira ou não. As sentenças sobre a lua, por exemplo, exigem que você conheça um pouco de astronomia rudimentar. Imagine uma criança pequena que se convenceu de que a lua é um nabo gigante. Ela poderia entender o que a sentença 'A lua é um nabo gigante' significa, mas erroneamente pensaria que era verdadeira.

Considere outro exemplo: Se M significa 'É manhã agora', então se é verdadeira ou não depende de quando você está lendo isso. Eu sei o que a sentença significa, mas - como não sei quando você estará lendo isso - não sei se é verdadeira ou falsa.

Portanto, uma interpretação sozinha não determina se uma sentença é verdadeira ou falsa. Verdade ou falsidade depende também de como o mundo é. Se M significasse 'A lua é um nabo gigante' e a lua real fosse um nabo gigante, então M seria verdadeira. Para colocar o ponto de forma geral, verdade ou falsidade é determinada por uma interpretação *mais* uma maneira como o mundo é.

INTERPRETAÇÃO + ESTADO DO MUNDO $\Longrightarrow$ VERDADE/FALSIDADE

Ao fornecer uma definição lógica de verdade, não seremos capazes de dar uma conta de como uma sentença atômica é tornada verdadeira ou falsa pelo mundo.

Em vez disso, introduziremos uma *atribuição de valor-verdade*. Formalmente, esta será uma função que nos diz o valor-verdade de todas as sentenças atômicas. Chame esta função de 'a' (para 'atribuição'). Definimos a para todas as letras de sentença $\mathcal{P}$, tal que

$$a(\mathcal{P}) = \begin{cases} 1 & \text{se } \mathcal{P} \text{ for verdadeira,} \\ 0 & \text{caso contrário.} \end{cases}$$

Isso significa que a pega qualquer sentença de LV e atribui a ela um um ou um zero; um se a sentença for verdadeira, zero se a sentença for falsa. Os detalhes da função a são determinados pelo significado das letras de sentença junto com o estado do mundo. Se D significa 'Está escuro lá fora', então $a(D) = 1$ à noite ou durante uma tempestade forte, enquanto $a(D) = 0$ em um dia claro.

Você pode pensar em a como sendo como uma linha de uma tabela-verdade. Enquanto uma linha de tabela-verdade atribui um valor-verdade a algumas sentenças atômicas, a atribuição de valor-verdade atribui um valor a cada sentença atômica de LV. Existem infinitas letras de sentença, e a atribuição de valor-verdade dá um valor a cada uma delas. Ao construir uma tabela-verdade, nos importamos apenas com letras de sentença que afetam o valor-verdade das sentenças que nos interessam. Como tal, ignoramos o resto. Estritamente falando, cada linha de uma tabela-verdade dá uma atribuição de valor-verdade *parcial*.

É importante notar que a atribuição de valor-verdade, a, não é parte da linguagem LV. Em vez disso, é parte da maquinaria matemática que estamos usando para descrever LV. Ela codifica quais sentenças atômicas são verdadeiras e quais são falsas.

Agora definimos a função verdade, v, usando a mesma estrutura recursiva que usamos para definir uma fbf de LV.

1. Se $\mathcal{A}$ é uma letra de sentença, então $v(\mathcal{A}) = a(\mathcal{A})$.
2. Se $\mathcal{A}$ é $\neg\mathcal{B}$ para alguma sentença $\mathcal{B}$, então

$$v(\mathcal{A}) = \begin{cases} 1 & \text{se } v(\mathcal{B}) = 0, \\ 0 & \text{caso contrário.} \end{cases}$$

3. Se $\mathcal{A}$ é $(\mathcal{B}\,\&\,\mathcal{C})$ para algumas sentenças $\mathcal{B}, \mathcal{C}$, então

$$v(\mathcal{A}) = \begin{cases} 1 & \text{se } v(\mathcal{B}) = 1 \text{ e } v(\mathcal{C}) = 1, \\ 0 & \text{caso contrário.} \end{cases}$$

Pode parecer que esta definição é circular, porque usa a palavra 'e' ao tentar definir 'e'. Note, no entanto, que esta não é uma definição da palavra em português 'e'; é uma definição de verdade para sentenças de LV contendo o símbolo lógico '&'. Definimos verdade para sentenças da linguagem objeto contendo o símbolo '&' usando a palavra da metalinguagem 'e'. Não há nada circular nisso.

4. Se $\mathcal{A}$ é $(\mathcal{B} \vee \mathcal{C})$ para algumas sentenças $\mathcal{B}, \mathcal{C}$, então

$$v(\mathcal{A}) = \begin{cases} 0 & \text{se } v(\mathcal{B}) = 0 \text{ e } v(\mathcal{C}) = 0, \\ 1 & \text{caso contrário.} \end{cases}$$

5. Se $\mathcal{A}$ é $(\mathcal{B} \rightarrow \mathcal{C})$ para algumas sentenças $\mathcal{B}, \mathcal{C}$, então

$$v(\mathcal{A}) = \begin{cases} 0 & \text{se } v(\mathcal{B}) = 1 \text{ e } v(\mathcal{C}) = 0, \\ 1 & \text{caso contrário.} \end{cases}$$

6. Se $\mathcal{A}$ é $(\mathcal{B} \leftrightarrow \mathcal{C})$ para algumas sentenças $\mathcal{B}, \mathcal{C}$, então

$$v(\mathcal{A}) = \begin{cases} 1 & \text{se } v(\mathcal{B}) = v(\mathcal{C}), \\ 0 & \text{caso contrário.} \end{cases}$$

Como a definição de v tem a mesma estrutura que a definição de uma fbf, sabemos que v atribui um valor a *toda* fbf de LV. Como as sentenças de LV e as fbfs de LV são as mesmas, isso significa que v retorna o valor-verdade de toda sentença de LV.

Verdade em LV é sempre verdade *relativa a* alguma atribuição de valor-verdade, porque a definição de verdade para LV não diz se uma dada sentença é verdadeira ou falsa. Em vez disso, diz como a verdade dessa sentença se relaciona com uma atribuição de valor-verdade.

Outros conceitos em LV

Trabalhando com LV até agora, passamos sem uma definição precisa de 'tautologia', 'contradição' e assim por diante. Tabelas-verdade forneciam uma maneira de *verificar se* uma sentença era uma tautologia em LV, mas elas não *definiam* o que significa ser uma tautologia em LV. Daremos definições desses conceitos para LV em termos de consequência.

A relação de consequência semântica, '$\mathcal{A}$ acarreta $\mathcal{B}$', significa que não há atribuição de valor-verdade para a qual $\mathcal{A}$ é verdadeira e $\mathcal{B}$ é falsa. Colocado de outra forma, significa que $\mathcal{B}$ é verdadeira para quaisquer e todas as atribuições de valor-verdade para as quais $\mathcal{A}$ é verdadeira.

Abreviamos isso com um símbolo chamado *duplo torniquete*: $\mathcal{A} \models \mathcal{B}$ significa '$\mathcal{A}$ acarreta semanticamente $\mathcal{B}$.'

Podemos falar sobre consequência entre mais de duas sentenças:

$$\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\} \models \mathcal{B}$$

significa que não há atribuição de valor-verdade para a qual todas as sentenças no conjunto $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ são verdadeiras e $\mathcal{B}$ é falsa.

Também podemos usar o símbolo com apenas uma sentença: $\models \mathcal{C}$ significa que $\mathcal{C}$ é verdadeira para todas as atribuições de valor-verdade. Isso é equivalente a dizer que a sentença é acarretada por qualquer coisa.

O símbolo de duplo torniquete nos permite dar definições concisas para vários conceitos de LV:

> Uma TAUTOLOGIA EM LV é uma sentença $\mathcal{A}$ tal que $\models \mathcal{A}$.
>
> Uma CONTRADIÇÃO EM LV é uma sentença $\mathcal{A}$ tal que $\models \neg\mathcal{A}$.
>
> Uma sentença é CONTINGENTE EM LV se e somente se não é nem uma tautologia nem uma contradição.
>
> Um argumento " $\mathcal{P}_1, \mathcal{P}_2, \cdots, \therefore\ \mathcal{C}$ " é VÁLIDO EM LV se e somente se $\{\mathcal{P}_1, \mathcal{P}_2, \cdots\} \models \mathcal{C}$.
>
> Duas sentenças $\mathcal{A}$ e $\mathcal{B}$ são LOGICAMENTE EQUIVALENTES EM LV se e somente se ambos $\mathcal{A} \models \mathcal{B}$ e $\mathcal{B} \models \mathcal{A}$.

A consistência lógica é um pouco mais difícil de definir em termos de consequência semântica. Em vez disso, vamos defini-la desta forma:

> O conjunto $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ é CONSISTENTE EM LV se e somente se há pelo menos uma atribuição de valor-verdade para a qual todas as sentenças são verdadeiras. O conjunto é INCONSISTENTE EM LV se e somente se não houver tal atribuição.

5.2 Interpretações e modelos em LQ

Em LV, uma interpretação ou chave de simbolização especifica o que cada uma das letras de sentença significa. A interpretação de uma letra de sentença junto com o estado do mundo determina se a letra de sentença é verdadeira ou falsa. Como as unidades básicas são letras de sentença, uma interpretação só importa na medida em que torna as letras de sentença verdadeiras ou falsas. Formalmente, a semântica para LV é estritamente em termos de atribuições de valor-verdade. Duas interpretações são as mesmas, formalmente, se elas resultam na mesma atribuição de valor-verdade.

O que é uma interpretação em LQ? Como uma chave de simbolização para LQ, uma interpretação requer um UD, um significado esquemático para cada um dos predicados e um objeto que é denotado por cada constante. Por exemplo:

UD: personagens de quadrinhos
Fx: x combate o crime.
b: o Batman
w: Bruce Wayne

Considere a sentença Fb. A sentença é verdadeira nesta interpretação, mas - assim como em LV - a sentença não é verdadeira *apenas por causa* da interpretação. A maioria das pessoas em nossa cultura sabe que o Batman combate o crime, mas isso requer um mínimo de conhecimento sobre quadrinhos. A sentença Fb é verdadeira por causa da interpretação *mais* alguns fatos sobre quadrinhos. Isso é especialmente óbvio quando consideramos Fw. Bruce Wayne é a identidade secreta do Batman nos quadrinhos - a afirmação de identidade $b = w$ é verdadeira - então Fw é verdadeira. Como é uma identidade *secreta*, no entanto, outros personagens não sabem que Fw é verdadeira, embora saibam que Fb é verdadeira.

Poderíamos tentar caracterizar isso como uma atribuição de valor-verdade, como fizemos para LV. A atribuição de valor-verdade atribuiria 0 ou 1 a cada fbf atômica: Fb, Fw e assim por diante. Se fôssemos fazer isso, no entanto, poderíamos igualmente traduzir as sentenças de LQ para LV substituindo Fb e Fw por letras de sentença. Poderíamos então contar com a definição de verdade para LV, mas ao custo de ignorar toda a estrutura lógica de predicados e termos. Ao escrever uma chave de simbolização para LQ, não damos definições separadas para Fb e Fw. Em vez disso, damos significados para F, b e w. Isso é essencial porque queremos ser capazes de usar quantificadores. Não há uma maneira adequada de traduzir $\forall x Fx$ em LV.

Então queremos uma contraparte formal para uma interpretação para predicados e constantes, não apenas para sentenças. Não podemos usar uma atribuição de valor-verdade para isso, porque um predicado não é nem verdadeiro nem falso. Na interpretação dada acima, F é verdadeiro *de* o Batman (ou seja, Fb é verdadeiro), mas não faz sentido algum perguntar se F sozinho é verdadeiro. Seria como perguntar se o fragmento da língua portuguesa '...combate o crime' é verdadeiro.

O que uma interpretação faz para um predicado, se não o torna verdadeiro ou falso? Uma interpretação ajuda a selecionar os objetos aos quais o predicado se aplica. Interpretar Fx como significando 'x combate o crime' seleciona Batman, Superman, Homem-Aranha e outros heróis como as coisas que são Fs. Formalmente, este é um conjunto de membros do UD aos quais o predicado se aplica; este conjunto é chamado de EXTENSÃO do predicado.

Muitos predicados têm extensões indefinidamente grandes. Seria impraticável tentar escrever todos os combatentes do crime dos quadrinhos individualmente, então em vez disso usamos uma expressão em português para interpretar o predicado. Isso é um tanto impreciso, porque a interpretação sozinha não diz quais membros do UD estão na extensão do predicado. Para descobrir se um membro particular do UD está na extensão do predicado (para descobrir se o Raio Negro combate o crime, por exemplo), você precisa saber sobre quadrinhos. Em geral, a extensão de um predicado é o resultado de uma interpretação *junto com* alguns fatos.

Às vezes é possível listar todas as coisas que estão na extensão de um predicado. Em vez de escrever uma sentença esquemática em português, podemos escrever a extensão como um conjunto de coisas. Suponha que quiséssemos adicionar um predicado de um lugar M à chave acima. Queremos que Mx signifique 'x mora na Mansão Wayne', então escrevemos a extensão como um conjunto de personagens:

extension(M) = {Bruce Wayne, Alfred the butler, Dick Grayson}

Você não precisa saber nada sobre quadrinhos para ser capaz de determinar que, nesta interpretação, Mw é verdadeira: Bruce Wayne é apenas especificado como uma das coisas que é M. Da mesma forma, $\exists x Mx$ é obviamente verdadeira nesta interpretação: Há pelo menos um membro do UD que é um M - de fato, há três deles.

E a sentença $\forall x Mx$? A sentença é falsa, porque não é verdade que todos os membros do UD são M. Requer o mínimo de conhecimento sobre quadrinhos para saber que há outros personagens além apenas desses três. Embora tenhamos especificado a extensão de M de uma maneira formalmente precisa, ainda especificamos o UD com uma descrição em português. Formalmente falando, um UD é apenas um conjunto de membros.

A significância formal de um predicado é determinada por sua extensão, mas o que devemos dizer sobre constantes como b e w? O significado de uma constante determina qual membro do UD é denotado pela constante. O indivíduo que a constante denota é chamado de REFERENTE da constante. Tanto b quanto w têm o mesmo referente, já que ambos se referem ao mesmo personagem de quadrinhos. Você pode pensar em uma letra constante como um nome e o referente como a coisa nomeada. Em português, podemos usar os diferentes nomes 'Batman' e 'Bruce Wayne' para nos referirmos ao mesmo personagem de quadrinhos. Nesta interpretação, podemos usar as diferentes constantes 'b' e 'w' para nos referirmos ao mesmo membro do UD.

Conjuntos

Usamos chaves '{' e '}' para denotar conjuntos. Os membros do conjunto podem ser listados em qualquer ordem, separados por vírgulas. O fato de os conjuntos poderem estar em qualquer ordem é importante, porque significa que {foo, bar} e {bar, foo} são o mesmo conjunto.

É possível ter um conjunto sem membros. Isso é chamado de CONJUNTO VAZIO. O conjunto vazio é às vezes escrito como {}, mas geralmente é escrito como o símbolo único $\emptyset$.

Modelos

Como vimos, uma interpretação em LQ é apenas formalmente significativa na medida em que determina um UD, uma extensão para cada predicado e um referente para cada constante. Chamamos essa estrutura formal de MODELO para LQ.

Para ver como isso funciona, considere esta chave de simbolização:

UD: Pessoas que atuaram como parte dos Três Patetas
Hx: x tinha cabelo na cabeça.
f: Senhor Fine

Se você não sabe nada sobre os Três Patetas, não será capaz de dizer quais sentenças de LQ são verdadeiras nesta interpretação. Talvez você apenas se lembre de Larry, Curly e Moe. A sentença Hf é verdadeira ou falsa? Depende de qual dos patetas é o Senhor Fine.

Qual é o modelo que corresponde a esta interpretação? Houve seis pessoas que atuaram como parte dos Três Patetas ao longo dos anos, então o UD terá seis membros: Larry Fine, Moe Howard, Curly Howard, Shemp Howard, Joe Besser e Curly Joe DeRita. Curly, Joe e Curly Joe foram os únicos patetas completamente carecas. O resultado é este modelo:

$$\begin{aligned} \text{UD} &= \{\text{Larry, Curly, Moe, Shemp, Joe, Curly Joe}\} \\ \text{extension}(H) &= \{\text{Larry, Moe, Shemp}\} \\ \text{referent}(f) &= \text{Larry} \end{aligned}$$

Você não precisa saber nada sobre os Três Patetas para avaliar se as sentenças são verdadeiras ou falsas neste *modelo*. Hf é verdadeira, já que o referente de f (Larry) está na extensão de H. Tanto $\exists xHx$ quanto $\exists x\neg Hx$ são verdadeiras, já que há pelo menos um membro do UD que está na extensão de H e pelo menos um membro que não está na extensão de H. Dessa forma, o modelo captura toda a significância formal da interpretação.

Agora considere esta interpretação:

UD: números inteiros menores que 10
Ex: x é par.
Nx: x é negativo.
Lxy: x é menor que y.
Txyz: x vezes y é igual a z.

Qual é o modelo que acompanha esta interpretação? O UD é o conjunto $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

A extensão de um predicado de um lugar como E ou N é apenas o subconjunto do UD do qual o predicado é verdadeiro. Grosso modo, a extensão do predicado E é o conjunto de Es no UD. A extensão de E é o subconjunto $\{2, 4, 6, 8\}$. Há muitos números pares além desses quatro, mas estes são os únicos membros do UD que são pares. Não há números negativos no UD, então N tem uma extensão vazia; ou seja, extension$(N) = \emptyset$.

A extensão de um predicado de dois lugares como L é um tanto complicada. Parece que a extensão de L deveria conter 1, já que 1 é menor que todos os outros números; deveria conter 2, já que 2 é menor que todos os outros números além de 1; e assim por diante. Todo membro do UD além de 9 é menor que algum membro do UD. O que aconteceria se simplesmente escrevêssemos extension(L) $= \{1, 2, 3, 4, 5, 6, 7, 8\}$?

O problema é que conjuntos podem ser escritos em qualquer ordem, então isso seria o mesmo que escrever extension(L) $= \{8, 7, 6, 5, 4, 3, 2, 1\}$. Isso não nos diz quais dos membros do conjunto são menores que quais outros membros.

Precisamos de alguma maneira de mostrar que 1 é menor que 8, mas que 8 não é menor que 1. A solução é ter a extensão de L consistindo de pares de números. Um PAR ORDENADO é como um conjunto com dois membros, exceto que a ordem *importa*. Escrevemos pares ordenados com colchetes angulares '<' e '>'. O par ordenado <foo, bar> é diferente do par ordenado <bar, foo>. A extensão de L é uma coleção de pares ordenados, todos os pares de números no UD tais que o primeiro número é menor que o segundo. Escrevendo isso completamente:

> extension(L) = {<1,2>, <1,3>, <1,4>, <1,5>, <1,6>, <1,7>, <1,8>, <1,9>, <2,3>, <2,4>, <2,5>, <2,6>, <2,7>, <2,8>, <2,9>, <3,4>, <3,5>, <3,6>, <3,7>, <3,8>, <3,9>, <4,5>, <4,6>, <4,7>, <4,8>, <4,9>, <5,6>, <5,7>, <5,8>, <5,9>, <6,7>, <6,8>, <6,9>, <7,8>, <7,9>, <8,9>}

Predicados de três lugares funcionarão de forma similar; a extensão de um predicado de três lugares é um conjunto de triplas ordenadas onde o predicado é verdadeiro dessas três coisas *nessa ordem.* Então a extensão de T neste modelo conterá triplas ordenadas como <2,4,8>, porque $2 \times 4 = 8$.

Geralmente, a extensão de um predicado de n-lugares é um conjunto de todas as n-uplas ordenadas $\langle a_1, a_2, \ldots, a_n \rangle$ tais que a_1–a_n são membros do UD e o predicado é verdadeiro de a_1–a_n naquela ordem.

5.3 Semântica para identidade

Identidade é um predicado especial de LQ. Escrevemos um pouco diferente de outros predicados de dois lugares: $x = y$ em vez de Ixy. Também não precisamos incluí-lo em uma chave de simbolização. A sentença $x = y$ sempre significa 'x é idêntico a y', e não pode ser interpretada para significar qualquer outra coisa. Da mesma forma, quando você constrói um modelo, você não pode escolher quais pares ordenados vão para a extensão do predicado de identidade. Ele sempre contém apenas o par ordenado de cada objeto no UD com si mesmo.

A sentença $\forall x Ixx$, que contém um predicado de dois lugares comum, é contingente. Se é verdadeira para uma interpretação depende de como você interpreta I, e se é verdadeira em um modelo depende da extensão de I.

A sentença $\forall x\ x = x$ é uma tautologia. A extensão da identidade sempre a tornará verdadeira.

Note que, embora a identidade sempre tenha a mesma interpretação, nem sempre tem a mesma extensão. A extensão da identidade depende do UD. Se o UD em um modelo é o conjunto {Doug}, então extension(=) naquele modelo é {<Doug, Doug>}. Se o UD é o conjunto {Doug, Omar}, então extension(=) naquele modelo é {<Doug, Doug>, <Omar, Omar>}. E assim por diante.

Se o referente de duas constantes é o mesmo, então qualquer coisa que é verdadeira de uma é verdadeira da outra. Por exemplo, se referent(a) = referent(b), então $Aa \leftrightarrow Ab$, $Ba \leftrightarrow Bb$, $Ca \leftrightarrow Cb$, $Rca \leftrightarrow Rcb$, $\forall xRxa \leftrightarrow \forall xRxb$, e assim por diante para quaisquer duas sentenças contendo a e b. No entanto, o inverso não é verdade.

É possível que qualquer coisa que seja verdadeira de a também seja verdadeira de b, mas ainda assim a e b tenham referentes diferentes. Isso pode parecer intrigante, mas é fácil construir um modelo que mostre isso. Considere este modelo:

$$\begin{aligned}
\text{UD} &= \{\text{Rosencrantz, Guildenstern}\}\\
\text{referent}(a) &= \text{Rosencrantz}\\
\text{referent}(b) &= \text{Guildenstern}\\
\text{para todos os predicados } \mathcal{P}\text{, extension}(\mathcal{P}) &= \emptyset\\
\text{extension}(=) &= \{<\text{Rosencrantz, Rosencrantz}>,\\
&\quad <\text{Guildenstern, Guildenstern}>\}
\end{aligned}$$

Isso especifica uma extensão para cada predicado de LQ: Todos os infinitos predicados estão vazios. Isso significa que tanto Aa quanto Ab são falsas, e elas são equivalentes; tanto Ba quanto Bb são falsas; e assim por diante para quaisquer duas sentenças que contenham a e b. No entanto, a e b se referem a coisas diferentes. Escrevemos a extensão da identidade para deixar isso claro: O par ordenado $\langle\text{referent}(a), \text{referent}(b)\rangle$ não está nela. Neste modelo, $a = b$ é falsa e $a \neq b$ é verdadeira.

5.4 Trabalhando com modelos

Usaremos o símbolo de duplo torniquete para LQ da mesma forma que fizemos para LV. '$\mathcal{A} \models \mathcal{B}$' significa que '$\mathcal{A}$ acarreta $\mathcal{B}$': Quando $\mathcal{A}$ e $\mathcal{B}$ são duas sentenças de LQ, $\mathcal{A} \models \mathcal{B}$ significa que não há modelo no qual $\mathcal{A}$ é verdadeira e $\mathcal{B}$ é falsa. $\models \mathcal{A}$ significa que $\mathcal{A}$ é verdadeira em todo modelo.

Isso nos permite dar definições para vários conceitos em LQ. Porque estamos usando o mesmo símbolo, estas definições parecerão semelhantes às definições em LV. Lembre-se, no entanto, de que as definições em LQ são em termos de *modelos* em vez de em termos de atribuições de valor-verdade.

> Uma TAUTOLOGIA EM LQ é uma sentença $\mathcal{A}$ que é verdadeira em todo modelo; ou seja, $\models \mathcal{A}$.
>
> Uma CONTRADIÇÃO EM LQ é uma sentença $\mathcal{A}$ que é falsa em todo modelo; ou seja, $\models \neg\mathcal{A}$.
>
> Uma sentença é CONTINGENTE EM LQ se e somente se não é nem uma tautologia nem uma contradição.
>
> Um argumento " $\mathcal{P}_1, \mathcal{P}_2, \cdots, \therefore \mathcal{C}$ " é VÁLIDO EM LQ se e somente se não há modelo no qual todas as premissas são verdadeiras e a conclusão é falsa; ou seja, $\{\mathcal{P}_1, \mathcal{P}_2, \cdots\} \models \mathcal{C}$. É INVÁLIDO EM LQ caso contrário.
>
> Duas sentenças $\mathcal{A}$ e $\mathcal{B}$ são LOGICAMENTE EQUIVALENTES EM LQ se e somente se ambos $\mathcal{A} \models \mathcal{B}$ e $\mathcal{B} \models \mathcal{A}$.
>
> O conjunto $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ é CONSISTENTE EM LQ se e somente se há pelo menos um modelo no qual todas as sentenças são verdadeiras. O conjunto é INCONSISTENTE EM LQ se e somente se não houver tal modelo.

Construindo modelos

Suponha que queiramos mostrar que $\forall x Axx \rightarrow Bd$ *não* é uma tautologia. Isso requer mostrar que a sentença não é verdadeira em todo modelo; ou seja, que é falsa em algum modelo. Se pudermos fornecer apenas um modelo no qual a sentença é falsa, então teremos mostrado que a sentença não é uma tautologia.

Como seria tal modelo? Para que $\forall x Axx \rightarrow Bd$ seja falsa, o antecedente ($\forall x Axx$) deve ser verdadeiro, e o consequente (Bd) deve ser falso.

Para construir tal modelo, começamos com um UD. Será mais fácil especificar extensões para predicados se tivermos um UD pequeno, então comece com um UD que tem apenas um membro. Formalmente, este único membro pode ser qualquer coisa. Digamos que é a cidade de Paris.

Queremos que $\forall x Axx$ seja verdadeiro, então queremos que todos os membros do UD sejam pareados com eles mesmos na extensão de A; isso significa que a extensão de A deve ser {<Paris,Paris>}.

Queremos que Bd seja falso, então o referente de d não deve estar na extensão de B. Damos a B uma extensão vazia.

Como Paris é o único membro do UD, deve ser o referente de d. O modelo que construímos se parece com isto:

$$\begin{aligned} \text{UD} &= \{\text{Paris}\} \\ \text{extension}(A) &= \{<\text{Paris,Paris}>\} \\ \text{extension}(B) &= \emptyset \\ \text{referent}(d) &= \text{Paris} \end{aligned}$$

Estritamente falando, um modelo especifica uma extensão para *cada* predicado de LQ e um referente para *cada* constante. Como tal, geralmente é impossível escrever um modelo completo. Isso exigiria escrever infinitas extensões e infinitos referentes. No entanto, não precisamos considerar todos os predicados para mostrar que há modelos nos quais $\forall xAxx \rightarrow Bd$ é falsa. Predicados como H e constantes como f_{13} não fazem diferença para a verdade ou falsidade desta sentença. É suficiente especificar extensões para A e B e um referente para d, como fizemos. Isso fornece um *modelo parcial* no qual a sentença é falsa.

Talvez você esteja se perguntando: O que o predicado A significa em português? O modelo parcial poderia corresponder a uma interpretação como esta:

UD: Paris
***Axy*:** x está no mesmo país que y.
***Bx*:** x foi fundada no século 20.
***d*:** a Cidade das Luzes

No entanto, tudo o que o modelo parcial nos diz é que A é um predicado que é verdadeiro de Paris e Paris. Há infinitos predicados em português que têm esta extensão. Axy poderia, em vez disso, traduzir 'x tem o mesmo tamanho que y' ou 'x e y são ambas cidades.' Da mesma forma, Bx é algum predicado que não se aplica a Paris; poderia, em vez disso, traduzir 'x está em uma ilha' ou 'x é um carro subcompacto.' Quando especificamos as extensões de A e B, não especificamos quais predicados em português A e B deveriam ser usados para traduzir. Estamos preocupados com se $\forall xAxx \rightarrow Bd$ resulta verdadeira ou falsa, e tudo o que importa para verdade e falsidade em LQ é a informação no modelo: o UD, as extensões dos predicados e os referentes das constantes.

Podemos igualmente facilmente mostrar que $\forall xAxx \rightarrow Bd$ não é uma contradição. Precisamos apenas especificar um modelo no qual $\forall xAxx \rightarrow Bd$ é verdadeira; ou seja, um modelo no qual ou $\forall xAxx$ é falso ou Bd é verdadeiro. Aqui está um modelo parcial:

$$\begin{aligned} \text{UD} &= \{\text{Paris}\} \\ \text{extension}(A) &= \{<\text{Paris,Paris}>\} \\ \text{extension}(B) &= \{\text{Paris}\} \\ \text{referent}(d) &= \text{Paris} \end{aligned}$$

Agora mostramos que $\forall xAxx \rightarrow Bd$ não é nem uma tautologia nem uma contradição. Pela definição de 'contingente em LQ', isso significa que $\forall xAxx \rightarrow Bd$ é contingente. Em geral, mostrar que uma sentença é contingente exigirá dois modelos: um no qual a sentença é verdadeira e outro no qual a sentença é falsa.

Suponha que queiramos mostrar que $\forall xSx$ e $\exists xSx$ não são logicamente equivalentes. Precisamos construir um modelo no qual as duas sentenças têm valores de verdade diferentes; queremos que uma delas seja verdadeira e a outra falsa. Começamos especificando um UD. Novamente, fazemos o UD pequeno para que possamos especificar extensões facilmente. Precisaremos de pelo menos dois membros. Seja o UD {Duke, Miles}. (Se escolhêssemos um UD com apenas um membro, as duas sentenças acabariam com o mesmo valor de verdade. Para ver o porquê, tente construir alguns modelos parciais com UDs de um membro.)

Podemos fazer $\exists x Sx$ verdadeira incluindo algo na extensão de S, e podemos fazer $\forall x Sx$ falsa deixando algo de fora da extensão de S. Não importa qual incluímos e qual deixamos de fora. Fazendo Duke o único S, obtemos um modelo parcial que se parece com isto:

$$\begin{aligned} \text{UD} &= \{\text{Duke, Miles}\} \\ \text{extension}(S) &= \{\text{Duke}\} \end{aligned}$$

Este modelo parcial mostra que as duas sentenças *não* são logicamente equivalentes.

Voltando à p. 65, dissemos que este argumento seria inválido em LQ:

$$\begin{array}{l} (Rc \,\&\, K_1c) \,\&\, Tc \\ \therefore\ Tc \,\&\, K_2c \end{array}$$

Para mostrar que é inválido, precisamos mostrar que há algum modelo no qual as premissas são verdadeiras e a conclusão é falsa. Podemos construir tal modelo deliberadamente. Aqui está uma maneira de fazê-lo:

$$\begin{aligned} \text{UD} &= \{\text{Björk}\} \\ \text{extension}(T) &= \{\text{Björk}\} \\ \text{extension}(K_1) &= \{\text{Björk}\} \\ \text{extension}(K_2) &= \emptyset \\ \text{extension}(R) &= \{\text{Björk}\} \\ \text{referent}(c) &= \text{Björk} \end{aligned}$$

Da mesma forma, podemos mostrar que um conjunto de sentenças é consistente construindo um modelo no qual todas as sentenças são verdadeiras.

Raciocinando sobre todos os modelos

Podemos mostrar que uma sentença *não* é uma tautologia apenas fornecendo um modelo cuidadosamente especificado: um modelo no qual a sentença é falsa. Para mostrar que algo é uma tautologia, por outro lado, não seria suficiente construir dez, cem ou mesmo mil modelos nos quais a sentença é verdadeira. Só é uma tautologia se for verdadeira em *todo* modelo, e há infinitos modelos. Isso não pode ser evitado apenas construindo modelos parciais, porque há infinitos modelos parciais.

Considere, por exemplo, a sentença $Raa \leftrightarrow Raa$. Há dois modelos parciais logicamente distintos desta sentença que têm um UD de 1 membro. Há 32 modelos parciais distintos que têm um UD de 2 membros. Há 1526 modelos parciais distintos que têm um UD de 3 membros. Há 262.144 modelos parciais distintos que têm um UD de 4 membros. E assim por diante até o infinito. Para mostrar que esta sentença é uma tautologia, precisamos mostrar algo sobre todos estes modelos. Não há esperança de fazê-lo lidando com eles um de cada vez.

No entanto, $Raa \leftrightarrow Raa$ é obviamente uma tautologia. Podemos prová-la com um argumento simples:

> Há dois tipos de modelos: aqueles nos quais $\langle \text{referent}(a), \text{referent}(a) \rangle$ está na extensão de R e aqueles nos quais não está. No primeiro tipo de modelo, Raa é verdadeira; pela tabela-verdade para o bicondicional, $Raa \leftrightarrow Raa$ também é verdadeira. No segundo tipo de modelo, Raa é falsa; isso torna $Raa \leftrightarrow Raa$ verdadeira. Como a sentença é verdadeira em ambos os tipos de modelo, e como todo modelo é um dos dois tipos, $Raa \leftrightarrow Raa$ é verdadeira em todo modelo. Portanto, é uma tautologia.

Este argumento é válido, é claro, e sua conclusão é verdadeira. No entanto, não é um argumento em LQ. Em vez disso, é um argumento em português *sobre* LQ; é um argumento na metalinguagem. Não há um procedimento formal para avaliar ou construir argumentos em linguagem natural como este. A imprecisão da linguagem natural é a própria razão pela qual começamos a pensar em linguagens formais.

Há mais dificuldades com esta abordagem.

Considere a sentença $\forall x(Rxx \rightarrow Rxx)$, outra tautologia óbvia. Pode ser tentador raciocinar desta forma: '$Rxx \rightarrow Rxx$ é verdadeira em todo modelo, então $\forall x(Rxx \rightarrow Rxx)$ deve ser verdadeira.' O problema é que $Rxx \rightarrow Rxx$ *não* é verdadeira em todo modelo. Não é uma sentença e, portanto, *nem* é verdadeira *nem* falsa. Ainda não temos o vocabulário para dizer o que queremos dizer sobre $Rxx \rightarrow Rxx$. Na próxima seção, introduzimos o conceito de *satisfação*; depois de fazê-lo, estaremos mais aptos a fornecer um argumento de que $\forall x(Rxx \rightarrow Rxx)$ é uma tautologia.

É necessário raciocinar sobre uma infinidade de modelos para mostrar que uma sentença é uma tautologia. Da mesma forma, é necessário raciocinar sobre uma infinidade de modelos para mostrar que uma sentença é uma contradição, que duas sentenças são equivalentes, que um conjunto de sentenças é inconsistente ou que um argumento é válido. Há outras coisas que podemos mostrar construindo cuidadosamente um modelo ou dois. A Tabela 5.1 resume quais coisas são quais.

Tabela 5.1: É relativamente fácil responder a uma questão se você pode fazê-lo construindo um modelo ou dois. É muito mais difícil se você precisa raciocinar sobre todos os modelos possíveis. Esta tabela mostra quando construir modelos é suficiente.

	SIM	NÃO
$\mathcal{A}$ é uma tautologia?	mostrar que $\mathcal{A}$ deve ser verdadeira em qualquer modelo	*construir um modelo* no qual $\mathcal{A}$ é falsa
$\mathcal{A}$ é uma contradição?	mostrar que $\mathcal{A}$ deve ser falsa em qualquer modelo	*construir um modelo* no qual $\mathcal{A}$ é verdadeira
$\mathcal{A}$ é contingente?	*construir dois modelos*, um no qual $\mathcal{A}$ é verdadeira e outro no qual $\mathcal{A}$ é falsa	ou mostrar que $\mathcal{A}$ é uma tautologia ou mostrar que $\mathcal{A}$ é uma contradição
$\mathcal{A}$ e $\mathcal{B}$ são equivalentes?	mostrar que $\mathcal{A}$ e $\mathcal{B}$ devem ter o mesmo valor-verdade em qualquer modelo	*construir um modelo* no qual $\mathcal{A}$ e $\mathcal{B}$ têm valores-verdade diferentes
O conjunto $\mathbb{A}$ é consistente?	*construir um modelo* no qual todas as sentenças em $\mathbb{A}$ são verdadeiras	mostrar que as sentenças não poderiam ser todas verdadeiras em qualquer modelo
O argumento '$\mathcal{P}$, $\therefore$ $\mathcal{C}$' é válido?	mostrar que qualquer modelo no qual $\mathcal{P}$ é verdadeira deve ser um modelo no qual $\mathcal{C}$ é verdadeira	*construir um modelo* no qual $\mathcal{P}$ é verdadeira e $\mathcal{C}$ é falsa

5.5 Verdade em LQ

Para LV, dividimos a definição de verdade em duas partes: uma atribuição de valor-verdade (a) para letras de sentença e uma função verdade (v) para todas as sentenças. A função verdade cobria a maneira como sentenças complexas poderiam ser construídas a partir de letras de sentença e conectivos.

Da mesma forma que verdade para LV é sempre *verdade dada uma atribuição de valor-verdade*, verdade para LQ é *verdade em um modelo*. A sentença atômica mais simples de LQ consiste em um predicado de um lugar seguido por uma constante, como Pj. É verdadeira em um modelo $\mathbb{M}$ se e somente se o referente de j está na extensão de P em $\mathbb{M}$.

Poderíamos continuar desta forma para definir verdade para todas as sentenças atômicas que contêm apenas predicados e constantes: Considere qualquer sentença da forma $\mathcal{R}c_1 \ldots c_n$ onde $\mathcal{R}$ é um predicado de n-lugares e os cs são constantes. É verdadeira em $\mathbb{M}$ se e somente se $\langle$referent$(c_1), \ldots,$ referent$(c_n)\rangle$ está em extension$(\mathcal{R})$ em $\mathbb{M}$.

Poderíamos então definir verdade para sentenças construídas com conectivos sentenciais da mesma forma que fizemos para LV. Por exemplo, a sentença $(Pj \rightarrow Mda)$ é verdadeira em $\mathbb{M}$ se Pj é falsa em $\mathbb{M}$ ou Mda é verdadeira em $\mathbb{M}$.

Infelizmente, esta abordagem falhará quando considerarmos sentenças contendo quantificadores. Considere $\forall x Px$. Quando é verdadeira em um modelo $\mathbb{M}$? A resposta não pode depender de se Px é verdadeira ou falsa em $\mathbb{M}$, porque o x em Px é uma variável livre. Px não é uma sentença. Não é nem verdadeira nem falsa.

Fomos capazes de dar uma definição recursiva de verdade para LV porque toda fórmula bem formada de LV tem um valor-verdade. Isso não é verdade em LQ, então não podemos definir verdade começando com a verdade de sentenças atômicas e construindo a partir daí. Também precisamos considerar as fórmulas atômicas que não são sentenças. Para fazer isso, definiremos *satisfação*; toda fórmula bem formada de LQ será satisfeita ou não, mesmo que não tenha um valor-verdade. Seremos então capazes de definir *verdade* para sentenças de LQ em termos de satisfação.

Satisfação

A fórmula Px diz, grosso modo, que x é um dos Ps. Isso não pode ser totalmente correto, no entanto, porque x é uma variável e não uma constante. Ela não nomeia nenhum membro particular do UD. Em vez disso, seu significado em uma sentença é determinado pelo quantificador que a liga. A variável x deve representar todos os membros do UD na sentença $\forall x Px$, mas só precisa representar um membro em $\exists x Px$. Como queremos que a definição de satisfação cubra Px sem qualquer quantificador, começaremos dizendo como interpretar uma variável livre como o x em Px.

Fazemos isso introduzindo uma *atribuição de variável.* Formalmente, esta é uma função que combina cada variável com um membro do UD. Chame esta função de 'a'. (O 'a' é para 'atribuição', mas esta não é a mesma que a atribuição de valor-verdade que usamos ao definir verdade para LV.)

A fórmula Px é satisfeita em um modelo $\mathbb{M}$ por uma atribuição de variável a se e somente se $a(x)$, o objeto que a atribui a x, está na extensão de P em $\mathbb{M}$.

Quando $\forall x Px$ é satisfeita? Não é suficiente se Px é satisfeita em $\mathbb{M}$ por a, porque isso apenas significa que $a(x)$ está em extension(P). $\forall x Px$ requer que todos os outros membros do UD estejam em extension(P) também.

Então precisamos de um pouco mais de notação técnica: Para qualquer membro Ω do UD e qualquer variável $\mathcal{X}$, seja $a[\Omega|\mathcal{X}]$ a atribuição de variável que atribui Ω a $\mathcal{X}$ mas concorda com a em todos os outros aspectos. Usamos Ω, a letra grega Ômega, para enfatizar que é algum membro do UD e não algum símbolo de LQ. Suponha, por exemplo, que o UD é presidentes dos Estados Unidos. A função $a[\text{Grover Cleveland}|x]$ atribui Grover Cleveland à variável x, independentemente do que a atribui a x; para qualquer outra variável, $a[\text{Grover Cleveland}|x]$ concorda com a.

Agora podemos dizer concisamente que $\forall x Px$ é satisfeita em um modelo $\mathbb{M}$ por uma atribuição de variável a se e somente se, para todo objeto Ω no UD de $\mathbb{M}$, Px é satisfeita em $\mathbb{M}$ por $a[\Omega|x]$.

Você pode se preocupar que isso seja circular, porque dá as condições de satisfação para a sentença $\forall x Px$ usando a frase 'para todo objeto'. No entanto, é importante lembrar a diferença entre um símbolo lógico como '$\forall$' e uma palavra em português como 'todo'. A palavra é parte da metalinguagem que usamos para definir condições de satisfação para sentenças da linguagem objeto que contêm o símbolo.

Agora podemos dar uma definição geral de satisfação, estendendo a partir dos casos que já discutimos. Definimos uma função s (para 'satisfação') em um modelo $\mathbb{M}$ tal que para qualquer fbf $\mathcal{A}$ e atribuição de variável a, $s(\mathcal{A}, a) = 1$ se $\mathcal{A}$ é satisfeita em $\mathbb{M}$ por a; caso contrário $s(\mathcal{A}, a) = 0$.

1. Se $\mathcal{A}$ é uma fbf atômica da forma $\mathcal{P}t_1 \ldots t_n$ e Ω_i é o objeto denotado por t_i, então

$$s(\mathcal{A}, a) = \begin{cases} 1 & \text{se } \langle \Omega_1 \ldots \Omega_n \rangle \text{ está em extension}(\mathcal{P}) \text{ em } \mathbb{M}, \\ 0 & \text{caso contrário.} \end{cases}$$

   Para cada termo t_i: Se t_i é uma constante, então $\Omega_i = \text{referent}(t_i)$. Se t_i é uma variável, então $\Omega_i = a(t_i)$.

2. Se $\mathcal{A}$ é $\neg\mathcal{B}$ para alguma fbf $\mathcal{B}$, então

$$s(\mathcal{A}, a) = \begin{cases} 1 & \text{se } s(\mathcal{B}, a) = 0, \\ 0 & \text{caso contrário.} \end{cases}$$

3. Se $\mathcal{A}$ é $(\mathcal{B} \,\&\, \mathcal{C})$ para algumas fbfs $\mathcal{B}, \mathcal{C}$, então

$$s(\mathcal{A}, a) = \begin{cases} 1 & \text{se } s(\mathcal{B}, a) = 1 \text{ e } s(\mathcal{C}, a) = 1, \\ 0 & \text{caso contrário.} \end{cases}$$

4. Se $\mathcal{A}$ é $(\mathcal{B} \vee \mathcal{C})$ para algumas fbfs $\mathcal{B}, \mathcal{C}$, então

$$s(\mathcal{A}, a) = \begin{cases} 0 & \text{se } s(\mathcal{B}, a) = 0 \text{ e } s(\mathcal{C}, a) = 0, \\ 1 & \text{caso contrário.} \end{cases}$$

5. Se $\mathcal{A}$ é $(\mathcal{B} \rightarrow \mathcal{C})$ para algumas fbfs $\mathcal{B}, \mathcal{C}$, então

$$s(\mathcal{A}, a) = \begin{cases} 0 & \text{se } s(\mathcal{B}, a) = 1 \text{ e } s(\mathcal{C}, a) = 0, \\ 1 & \text{caso contrário.} \end{cases}$$

6. Se $\mathcal{A}$ é $(\mathcal{B} \leftrightarrow \mathcal{C})$ para algumas sentenças $\mathcal{B}, \mathcal{C}$, então

$$s(\mathcal{A}, a) = \begin{cases} 1 & \text{se } s(\mathcal{B}, a) = s(\mathcal{C}, a), \\ 0 & \text{caso contrário.} \end{cases}$$

7. Se $\mathcal{A}$ é $\forall \chi \mathcal{B}$ para alguma fbf $\mathcal{B}$ e alguma variável χ, então

$$s(\mathcal{A}, a) = \begin{cases} 1 & \text{se } s(\mathcal{B}, a[\Omega|\chi]) = 1 \text{ para todo membro } \Omega \text{ do UD,} \\ 0 & \text{caso contrário.} \end{cases}$$

8. Se $\mathcal{A}$ é $\exists \chi \mathcal{B}$ para alguma fbf $\mathcal{B}$ e alguma variável χ, então

$$s(\mathcal{A}, a) = \begin{cases} 1 & \text{se } s(\mathcal{B}, a[\Omega|\chi]) = 1 \text{ para pelo menos um membro } \Omega \text{ do UD,} \\ 0 & \text{caso contrário.} \end{cases}$$

Esta definição segue a mesma estrutura que a definição de uma fbf para LQ, então sabemos que toda fbf de LQ será coberta por esta definição. Para um modelo $\mathbb{M}$ e uma atribuição de variável a, qualquer fbf será satisfeita ou não. Nenhuma fbf é deixada de fora ou atribuída valores conflitantes.

Verdade

Considere uma sentença simples como $\forall x Px$. Pela parte 7 na definição de satisfação, esta sentença é satisfeita se $a[\Omega|x]$ satisfaz Px em $\mathbb{M}$ para todo Ω no UD. Pela parte 1 da definição, este será o caso se todo Ω estiver na extensão de P. Se $\forall x Px$ é satisfeita não depende da atribuição de variável particular a. Se esta sentença é satisfeita, então é verdadeira. Esta é uma formalização do que dissemos o tempo todo: $\forall x Px$ é verdadeira se tudo no UD está na extensão de P.

A mesma coisa vale para qualquer sentença de LQ. Como todas as variáveis são ligadas, uma sentença é satisfeita ou não independentemente dos detalhes da atribuição de variável. Então podemos definir verdade desta forma: Uma sentença $\mathcal{A}$ é VERDADEIRA EM $\mathbb{M}$ se e somente se alguma atribuição de variável satisfaz $\mathcal{A}$ em M; $\mathcal{A}$ é FALSA EM $\mathbb{M}$ caso contrário.

Verdade em LQ é *verdade em um modelo*. Sentenças de LQ não são verdadeiras ou falsas de forma direta como meros símbolos, mas apenas relativamente a um modelo. Um modelo fornece o significado dos símbolos, na medida em que faz diferença para verdade e falsidade.

Raciocinando sobre todos os modelos (reprise)

No final da seção 5.4, ficamos impedidos quando tentamos mostrar que $\forall x(Rxx \rightarrow Rxx)$ é uma tautologia. Tendo definido satisfação, podemos agora raciocinar desta forma:

> Considere um modelo arbitrário $\mathbb{M}$. Agora considere um membro arbitrário do UD; por conveniência, chame-o de Ω. Deve ser o caso ou que $\langle\Omega, \Omega\rangle$ está na extensão de R ou que não está. Se $\langle\Omega, \Omega\rangle$ está na extensão de R, então Rxx é satisfeita por uma atribuição de variável que atribui Ω a x (pela parte 1 da definição de satisfação); como o consequente de $Rxx \rightarrow Rxx$ é satisfeito, o condicional é satisfeito (pela parte 5). Se $\langle\Omega, \Omega\rangle$ não está na extensão de R, então Rxx não é satisfeita por uma atribuição de variável que atribui Ω a x (pela parte 1); como o antecedente de $Rxx \rightarrow Rxx$ não é satisfeito, o condicional é satisfeito (pela parte 5). Em ambos os casos, $Rxx \rightarrow Rxx$ é satisfeita. Isso é verdade para qualquer membro do UD, então $\forall x(Rxx \rightarrow Rxx)$ é satisfeita por qualquer atribuição de valor-verdade (pela parte 7). Então $\forall x(Rxx \rightarrow Rxx)$ é verdadeira em $\mathbb{M}$ (pela definição de verdade). Este argumento vale independentemente do UD exato e independentemente da extensão exata de R, então $\forall x(Rxx \rightarrow Rxx)$ é verdadeira em qualquer modelo. Portanto, é uma tautologia.

Dar argumentos sobre todos os modelos possíveis normalmente requer uma combinação inteligente de duas estratégias:

1. Dividir casos entre dois tipos possíveis, tal que todo caso deve ser de um tipo ou do outro. No argumento na p. 95, por exemplo, distinguimos dois tipos de modelos com base em se um par ordenado específico estava ou não em extension(R). No argumento acima, distinguimos casos em que um par ordenado estava em extension(R) e casos em que não estava.

2. Considerar um objeto arbitrário como uma maneira de mostrar algo mais geral. No argumento acima, foi crucial que Ω fosse apenas algum membro arbitrário do UD. Não assumimos nada especial sobre ele. Como tal, qualquer coisa que pudéssemos mostrar valer para Ω deve valer para todo membro do UD - se pudéssemos mostrar para Ω, poderíamos mostrar para qualquer coisa. Da mesma forma, não assumimos nada especial sobre $\mathbb{M}$, e então qualquer coisa que pudéssemos mostrar sobre $\mathbb{M}$ deve valer para todos os modelos.

Considere mais um exemplo. O argumento $\forall x(Hx \,\&\, Jx) \therefore \forall x Hx$ é obviamente válido. Só podemos mostrar que o argumento é válido considerando o que deve ser verdade em todo modelo no qual a premissa é verdadeira.

> Considere um modelo arbitrário $\mathbb{M}$ no qual a premissa $\forall x(Hx \,\&\, Jx)$ é verdadeira. A conjunção $Hx \,\&\, Jx$ é satisfeita independentemente

> do que é atribuído a x, então Hx também deve ser (pela parte 3 da definição de satisfação). Como tal, $(\forall x)Hx$ é satisfeita por qualquer atribuição de variável (pela parte 7 da definição de satisfação) e verdadeira em $\mathbb{M}$ (pela definição de verdade). Como não assumimos nada sobre $\mathbb{M}$ além de $\forall x(Hx \,\&\, Jx)$ ser verdadeira, $(\forall x)Hx$ deve ser verdadeira em qualquer modelo no qual $\forall x(Hx \,\&\, Jx)$ é verdadeira. Então $\forall x(Hx \,\&\, Jx) \models \forall xHx$.

Mesmo para um argumento simples como este, o raciocínio é um tanto complicado. Para argumentos mais longos, o raciocínio pode ser insuportável. O problema surge porque falar sobre uma infinidade de modelos requer raciocinar em português. O que devemos fazer?

Poderíamos tentar formalizar nosso raciocínio sobre modelos, codificando as estratégias de dividir para conquistar que usamos acima. Esta abordagem, originalmente chamada *tableaux semânticos*, foi desenvolvida na década de 1950 por Evert Beth e Jaakko Hintikka. Seus tableaux são agora mais comumente chamados de *árvores de verdade*.

Uma abordagem mais tradicional é considerar argumentos dedutivos como provas. Um *sistema de prova* consiste em regras que distinguem formalmente entre argumentos legítimos e ilegítimos - sem considerar modelos ou os significados dos símbolos. No próximo capítulo, desenvolvemos sistemas de prova para LV e LQ.

Practice Exercises

$\star$ **Part A** Determine se cada sentença é verdadeira ou falsa no modelo dado.

$$\begin{aligned} \text{UD} &= \{\text{Corwin, Benedict}\} \\ \text{extension}(A) &= \{\text{Corwin, Benedict}\} \\ \text{extension}(B) &= \{\text{Benedict}\} \\ \text{extension}(N) &= \emptyset \\ \text{referent}(c) &= \text{Corwin} \end{aligned}$$

1. Bc
2. $Ac \leftrightarrow \neg Nc$
3. $Nc \rightarrow (Ac \vee Bc)$
4. $\forall xAx$
5. $\forall x\neg Bx$
6. $\exists x(Ax \,\&\, Bx)$
7. $\exists x(Ax \rightarrow Nx)$
8. $\forall x(Nx \vee \neg Nx)$
9. $\exists xBx \rightarrow \forall xAx$

⋆ **Part B** Determine se cada sentença é verdadeira ou falsa no modelo dado.

$$\begin{aligned}
\text{UD} &= \{\text{Waylan, Willy, Johnny}\}\\
\text{extension}(H) &= \{\text{Waylan, Willy, Johnny}\}\\
\text{extension}(W) &= \{\text{Waylan, Willy}\}\\
\text{extension}(R) &= \{<\text{Waylan, Willy}>,<\text{Willy, Johnny}>,<\text{Johnny, Waylan}>\}\\
\text{referent}(m) &= \text{Johnny}
\end{aligned}$$

1. $\exists x(Rxm \,\&\, Rmx)$
2. $\forall x(Rxm \vee Rmx)$
3. $\forall x(Hx \leftrightarrow Wx)$
4. $\forall x(Rxm \rightarrow Wx)$
5. $\forall x\big[Wx \rightarrow (Hx \,\&\, Wx)\big]$
6. $\exists x Rxx$
7. $\exists x \exists y Rxy$
8. $\forall x \forall y Rxy$
9. $\forall x \forall y(Rxy \vee Ryx)$
10. $\forall x \forall y \forall z\big[(Rxy \,\&\, Ryz) \rightarrow Rxz\big]$

Part C Determine se cada sentença é verdadeira ou falsa no modelo dado.

$$\begin{aligned}
\text{UD} &= \{\text{Lemmy, Courtney, Eddy}\}\\
\text{extension}(G) &= \{\text{Lemmy, Courtney, Eddy}\}\\
\text{extension}(H) &= \{\text{Courtney}\}\\
\text{extension}(M) &= \{\text{Lemmy, Eddy}\}\\
\text{referent}(c) &= \text{Courtney}\\
\text{referent}(e) &= \text{Eddy}
\end{aligned}$$

1. Hc
2. He
3. $Mc \vee Me$
4. $Gc \vee \neg Gc$
5. $Mc \rightarrow Gc$
6. $\exists x Hx$
7. $\forall x Hx$
8. $\exists x \neg Mx$
9. $\exists x(Hx \,\&\, Gx)$
10. $\exists x(Mx \,\&\, Gx)$
11. $\forall x(Hx \vee Mx)$
12. $\exists x Hx \,\&\, \exists x Mx$
13. $\forall x(Hx \leftrightarrow \neg Mx)$
14. $\exists x Gx \,\&\, \exists x \neg Gx$
15. $\forall x \exists y(Gx \,\&\, Hy)$

⋆ **Part D** Escreva o modelo que corresponde à interpretação dada.

UD: números naturais de 10 a 13
Ox: x é ímpar.
Sx: x é menor que 7.
Tx: x é um número de dois dígitos.
Ux: x é considerado azarado.
Nxy: x é o próximo número após y.

Part E Mostre que cada um dos seguintes é contingente.

⋆ 1. $Da \,\&\, Db$
⋆ 2. $\exists xTxh$
⋆ 3. $Pm \,\&\, \neg\forall xPx$
4. $\forall zJz \leftrightarrow \exists yJy$
5. $\forall x(Wxmn \vee \exists yLxy)$
6. $\exists x(Gx \rightarrow \forall yMy)$

⋆ **Part F** Mostre que os seguintes pares de sentenças não são logicamente equivalentes.

1. Ja, Ka
2. $\exists xJx$, Jm
3. $\forall xRxx$, $\exists xRxx$
4. $\exists xPx \rightarrow Qc$, $\exists x(Px \rightarrow Qc)$
5. $\forall x(Px \rightarrow \neg Qx)$, $\exists x(Px \,\&\, \neg Qx)$
6. $\exists x(Px \,\&\, Qx)$, $\exists x(Px \rightarrow Qx)$
7. $\forall x(Px \rightarrow Qx)$, $\forall x(Px \,\&\, Qx)$
8. $\forall x\exists yRxy$, $\exists x\forall yRxy$
9. $\forall x\exists yRxy$, $\forall x\exists yRyx$

Part G Mostre que os seguintes conjuntos de sentenças são consistentes.

1. {Ma, ¬Na, Pa, ¬Qa}
2. $\{Lee, Lef, \neg Lfe, \neg Lff\}$
3. $\{\neg(Ma \,\&\, \exists xAx), Ma \vee Fa, \forall x(Fx \rightarrow Ax)\}$
4. $\{Ma \vee Mb, Ma \rightarrow \forall x\neg Mx\}$
5. $\{\forall yGy, \forall x(Gx \rightarrow Hx), \exists y\neg Iy\}$
6. $\{\exists x(Bx \vee Ax), \forall x\neg Cx, \forall x\big[(Ax \,\&\, Bx) \rightarrow Cx\big]\}$
7. $\{\exists xXx, \exists xYx, \forall x(Xx \leftrightarrow \neg Yx)\}$
8. $\{\forall x(Px \vee Qx), \exists x\neg(Qx \,\&\, Px)\}$
9. $\{\exists z(Nz \,\&\, Ozz), \forall x\forall y(Oxy \rightarrow Oyx)\}$
10. $\{\neg\exists x\forall yRxy, \forall x\exists yRxy\}$

Part H Construa modelos para mostrar que os seguintes argumentos são inválidos.

1. $\forall x(Ax \to Bx)$, $\therefore\ \exists xBx$
2. $\forall x(Rx \to Dx)$, $\forall x(Rx \to Fx)$, $\therefore\ \exists x(Dx \,\&\, Fx)$
3. $\exists x(Px \to Qx)$, $\therefore \exists xPx$
4. $Na \,\&\, Nb \,\&\, Nc$, $\therefore\ \forall xNx$
5. Rde, $\exists xRxd$, $\therefore\ Red$
6. $\exists x(Ex \,\&\, Fx)$, $\exists xFx \to \exists xGx$, $\therefore\ \exists x(Ex \,\&\, Gx)$
7. $\forall xOxc$, $\forall xOcx$, $\therefore\ \forall xOxx$
8. $\exists x(Jx \,\&\, Kx)$, $\exists x\neg Kx$, $\exists x\neg Jx$, $\therefore\ \exists x(\neg Jx \,\&\, \neg Kx)$
9. $Lab \to \forall xLxb$, $\exists xLxb$, $\therefore\ Lbb$

Part I

⋆ 1. Mostre que $\{\neg Raa, \forall x(x = a \vee Rxa)\}$ é consistente.
⋆ 2. Mostre que $\{\forall x\forall y\forall z(x = y \vee y = z \vee x = z), \exists x\exists y\ x \neq y\}$ é consistente.
⋆ 3. Mostre que $\{\forall x\forall y\ x = y, \exists x\ x \neq a\}$ é inconsistente.
4. Mostre que $\exists x(x = h \,\&\, x = i)$ é contingente.
5. Mostre que $\{\exists x\exists y(Zx \,\&\, Zy \,\&\, x = y)$, $\neg Zd$, $d = s\}$ é consistente.
6. Mostre que '$\forall x(Dx \to \exists yTyx)$ $\therefore\ \exists y\exists z\ y \neq z$' é inválido.

Part J

1. Muitos livros de lógica definem consistência e inconsistência desta forma: " Um conjunto $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ é inconsistente se e somente se $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\} \models (\mathcal{B} \,\&\, \neg\mathcal{B})$ para alguma sentença $\mathcal{B}$. Um conjunto é consistente se não for inconsistente."

   Esta definição leva a quaisquer conjuntos diferentes sendo consistentes do que a definição na p. 86? Explique sua resposta.
⋆ 2. Nossa definição de verdade diz que uma sentença $\mathcal{A}$ é VERDADEIRA EM $\mathbb{M}$ se e somente se alguma atribuição de variável satisfaz $\mathcal{A}$ em M. Faria alguma diferença se disséssemos, em vez disso, que $\mathcal{A}$ é VERDADEIRA EM $\mathbb{M}$ se e somente se *toda* atribuição de variável satisfaz $\mathcal{A}$ em M? Explique sua resposta.

Chapter 6

Proofs

Consider two arguments in SL:

Argument A	Argument B
$P \vee Q$	$P \to Q$
$\neg P$	P
$\therefore$ Q	$\therefore$ Q

Clearly, these are valid arguments. You can confirm that they are valid by constructing four-line truth tables. Argument A makes use of an inference form that is always valid: Given a disjunction and the negation of one of the disjuncts, the other disjunct follows as a valid consequence. This rule is called *disjunctive syllogism.*

Argument B makes use of a different valid form: Given a conditional and its antecedent, the consequent follows as a valid consequence. This is called *modus ponens.*

When we construct truth tables, we do not need to give names to different inference forms. There is no reason to distinguish modus ponens from a disjunctive syllogism. For this same reason, however, the method of truth tables does not clearly show *why* an argument is valid. If you were to do a 1024-line truth table for an argument that contains ten sentence letters, then you could check to see if there were any lines on which the premises were all true and the conclusion were false. If you did not see such a line and provided you made no mistakes in constructing the table, then you would know that the argument was valid. Yet you would not be able to say anything further about why this particular argument was a valid argument form.

The aim of a *proof system* is to show that particular arguments are valid in a way that allows us to understand the reasoning involved in the argument. We begin with basic argument forms, like disjunctive syllogism and modus ponens. These forms can then be combined to make more complicated arguments, like this one:

(1) $\neg L \rightarrow (J \vee L)$
(2) $\neg L$
$\therefore$ J

By modus ponens, (1) and (2) entail $J \vee L$. This is an *intermediate conclusion*. It follows logically from the premises, but it is not the conclusion we want. Now $J \vee L$ and (2) entail J, by disjunctive syllogism. We do not need a new rule for this argument. The proof of the argument shows that it is really just a combination of rules we have already introduced.

Formally, a PROOF is a sequence of sentences. The first sentences of the sequence are assumptions; these are the premises of the argument. Every sentence later in the sequence follows from earlier sentences by one of the rules of proof. The final sentence of the sequence is the conclusion of the argument.

This chapter begins with a proof system for SL, which is then extended to cover QL and QL plus identity.

6.1 Basic rules for SL

In designing a proof system, we could just start with disjunctive syllogism and modus ponens. Whenever we discovered a valid argument which could not be proven with rules we already had, we could introduce new rules. Proceeding in this way, we would have an unsystematic grab bag of rules. We might accidently add some strange rules, and we would surely end up with more rules than we need.

Instead, we will develop what is called a NATURAL DEDUCTION system. In a natural deduction system, there will be two rules for each logical operator: an INTRODUCTION rule that allows us to prove a sentence that has it as the main logical operator and an ELIMINATION rule that allows us to prove something given a sentence that has it as the main logical operator.

In addition to the rules for each logical operator, we will also have a reiteration rule. If you already have shown something in the course of a proof, the reiteration rule allows you to repeat it on a new line. For instance:

1	$\mathcal{A}$	
2	$\mathcal{A}$	R 1

When we add a line to a proof, we write the rule that justifies that line. We also write the numbers of the lines to which the rule was applied. The reiteration rule above is justified by one line, the line that you are reiterating. So the 'R 1' on line 2 of the proof means that the line is justified by the reiteration rule (R) applied to line 1.

Obviously, the reiteration rule will not allow us to show anything *new*. For that, we will need more rules. The remainder of this section will give introduction and elimination rules for all of the sentential connectives. This will give us a complete proof system for SL. Later in the chapter, we introduce rules for quantifiers and identity.

All of the rules introduced in this chapter are summarized starting on p. 154.

Conjunction

Think for a moment: What would you need to show in order to prove $E \,\&\, F$?

Of course, you could show $E \,\&\, F$ by proving E and separately proving F. This holds even if the two conjuncts are not atomic sentences. If you can prove $[(A \vee J) \rightarrow V]$ and $[(V \rightarrow L) \leftrightarrow (F \vee N)]$, then you have effectively proven

$$[(A \vee J) \rightarrow V] \,\&\, [(V \rightarrow L) \leftrightarrow (F \vee N)].$$

So this will be our conjunction introduction rule, which we abbreviate &I:

m	$\mathcal{A}$	
n	$\mathcal{B}$	
	$\mathcal{A} \,\&\, \mathcal{B}$	&I m, n

A line of proof must be justified by some rule, and here we have '&I m,n.' This means: Conjunction introduction applied to line m and line n. These are variables, not real line numbers; m is some line and n is some other line. In an actual proof, the lines are numbered $1, 2, 3, \ldots$ and rules must be applied to specific line numbers. When we define the rule, however, we use variables to underscore the point that the rule may be applied to any two lines that are already in the proof. If you have K on line 8 and L on line 15, you can prove $(K \,\&\, L)$ at some later point in the proof with the justification '&I 8, 15.'

Now, consider the elimination rule for conjunction. What are you entitled to conclude from a sentence like $E \,\&\, F$? Surely, you are entitled to conclude E; if $E \,\&\, F$ were true, then E would be true. Similarly, you are entitled to conclude F. This will be our conjunction elimination rule, which we abbreviate &E:

$$
\begin{array}{r|ll}
m & \mathcal{A} \,\&\, \mathcal{B} & \\
 & \mathcal{A} & \&\text{E } m \\
 & \mathcal{B} & \&\text{E } m
\end{array}
$$

When you have a conjunction on some line of a proof, you can use &E to derive either of the conjuncts. The &E rule requires only one sentence, so we write one line number as the justification for applying it.

Even with just these two rules, we can provide some proofs. Consider this argument.

$$
\begin{array}{rl}
 & [(A \vee B) \to (C \vee D)] \,\&\, [(E \vee F) \to (G \vee H)] \\
\therefore & [(E \vee F) \to (G \vee H)] \,\&\, [(A \vee B) \to (C \vee D)]
\end{array}
$$

The main logical operator in both the premise and conclusion is conjunction. Since conjunction is symmetric, the argument is obviously valid. In order to provide a proof, we begin by writing down the premise. After the premises, we draw a horizontal line— everything below this line must be justified by a rule of proof. So the beginning of the proof looks like this:

$$
\begin{array}{r|l}
1 & [(A \vee B) \to (C \vee D)] \,\&\, [(E \vee F) \to (G \vee H)] \\ \hline
\end{array}
$$

From the premise, we can get each of the conjuncts by &E. The proof now looks like this:

$$
\begin{array}{r|ll}
1 & [(A \vee B) \to (C \vee D)] \,\&\, [(E \vee F) \to (G \vee H)] & \\ \hline
2 & [(A \vee B) \to (C \vee D)] & \&\text{E } 1 \\
3 & [(E \vee F) \to (G \vee H)] & \&\text{E } 1
\end{array}
$$

The rule &I requires that we have each of the conjuncts available somewhere in the proof. They can be separated from one another, and they can appear in any order. So by applying the &I rule to lines 3 and 2, we arrive at the desired conclusion. The finished proof looks like this:

$$
\begin{array}{r|ll}
1 & [(A \vee B) \to (C \vee D)] \,\&\, [(E \vee F) \to (G \vee H)] & \\ \hline
2 & [(A \vee B) \to (C \vee D)] & \&\text{E } 1 \\
3 & [(E \vee F) \to (G \vee H)] & \&\text{E } 1 \\
4 & [(E \vee F) \to (G \vee H)] \,\&\, [(A \vee B) \to (C \vee D)] & \&\text{I } 3, 2
\end{array}
$$

This proof is trivial, but it shows how we can use rules of proof together to demonstrate the validity of an argument form. Also: Using a truth table to show that this argument is valid would have required a staggering 256 lines, since there are eight sentence letters in the argument.

Disjunction

If M were true, then $M \vee N$ would also be true. So the disjunction introduction rule ($\vee$I) allows us to derive a disjunction if we have one of the two disjuncts:

m	$\mathcal{A}$	
	$\mathcal{A} \vee \mathcal{B}$	$\vee$I m
	$\mathcal{B} \vee \mathcal{A}$	$\vee$I m

Notice that $\mathcal{B}$ can be *any* sentence whatsoever. So the following is a legitimate proof:

1	M	
2	$M \vee ([(A \leftrightarrow B) \to (C \,\&\, D)] \leftrightarrow [E \,\&\, F])$	$\vee$I 1

It may seem odd that just by knowing M we can derive a conclusion that includes sentences like A, B, and the rest— sentences that have nothing to do with M. Yet the conclusion follows immediately by $\vee$I. This is as it should be: The truth conditions for the disjunction mean that, if $\mathcal{A}$ is true, then $\mathcal{A} \vee \mathcal{B}$ is true regardless of what $\mathcal{B}$ is. So the conclusion could not be false if the premise were true; the argument is valid.

Now consider the disjunction elimination rule. What can you conclude from $M \vee N$? You cannot conclude M. It might be M's truth that makes $M \vee N$ true, as in the example above, but it might not. From $M \vee N$ alone, you cannot conclude anything about either M or N specifically. If you also knew that N was false, however, then you would be able to conclude M.

This is just disjunctive syllogism, it will be the disjunction elimination rule ($\vee$E).

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\mathcal{A}$	$\vee$E m, n

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{A}$	
	$\mathcal{B}$	$\vee$E m, n

Conditional

Consider this argument:

$$R \vee F$$
$$\therefore\ \neg R \to F$$

The argument is certainly a valid one. What should the conditional introduction rule be, such that we can draw this conclusion?

We begin the proof by writing down the premise of the argument and drawing a horizontal line, like this:

$$\begin{array}{r|l}
1 & \underline{R \vee F}
\end{array}$$

If we had $\neg R$ as a further premise, we could derive F by the $\vee$E rule. We do not have $\neg R$ as a premise of this argument, nor can we derive it directly from the premise we do have— so we cannot simply prove F. What we will do instead is start a *subproof*, a proof within the main proof. When we start a subproof, we draw another vertical line to indicate that we are no longer in the main proof. Then we write in an assumption for the subproof. This can be anything we want. Here, it will be helpful to assume $\neg R$. Our proof now looks like this:

$$\begin{array}{r|l}
1 & \underline{R \vee F} \\
2 & \quad \begin{array}{|l} \underline{\neg R} \end{array}
\end{array}$$

It is important to notice that we are not claiming to have proven $\neg R$. We do not need to write in any justification for the assumption line of a subproof. You can think of the subproof as posing the question: What could we show *if* $\neg R$ were true? For one thing, we can derive F. So we do:

$$\begin{array}{r|lll}
1 & \underline{R \vee F} & & \\
2 & \quad \begin{array}{|l} \underline{\neg R} \end{array} & & \\
3 & \quad \begin{array}{|l} F \end{array} & & \vee\text{E } 1, 2
\end{array}$$

This has shown that *if* we had $\neg R$ as a premise, *then* we could prove F. In effect, we have proven $\neg R \rightarrow F$. So the conditional introduction rule ($\rightarrow$I) will allow us to close the subproof and derive $\neg R \rightarrow F$ in the main proof. Our final proof looks like this:

$$\begin{array}{r|lll}
1 & \underline{R \vee F} & & \\
2 & \quad \begin{array}{|l} \underline{\neg R} \end{array} & & \\
3 & \quad \begin{array}{|l} F \end{array} & & \vee\text{E } 1, 2 \\
4 & \neg R \rightarrow F & & \rightarrow\text{I } 2\text{–}3
\end{array}$$

Notice that the justification for applying the $\rightarrow$I rule is the entire subproof. Usually that will be more than just two lines.

It may seem as if the ability to assume anything at all in a subproof would lead to chaos: Does it allow you to prove any conclusion from any premises? The answer is no, it does not. Consider this proof:

$$\begin{array}{r|l}
1 & \mathcal{A} \\ \hline
2 & \quad \begin{array}{|l} \mathcal{B} \\ \hline \mathcal{B} \qquad \text{R } 2 \end{array} \\
3 &
\end{array}$$

It may seem as if this is a proof that you can derive any conclusions $\mathcal{B}$ from any premise $\mathcal{A}$. When the vertical line for the subproof ends, the subproof is *closed*. In order to complete a proof, you must close all of the subproofs. And you cannot close the subproof and use the R rule again on line 4 to derive $\mathcal{B}$ in the main proof. Once you close a subproof, you cannot refer back to individual lines inside it.

Closing a subproof is called *discharging* the assumptions of that subproof. So we can put the point this way: You cannot complete a proof until you have discharged all of the assumptions besides the original premises of the argument.

Of course, it is legitimate to do this:

1	$\mathcal{A}$	
2	$\quad \mathcal{B}$	
3	$\quad \mathcal{B}$	R 2
4	$\mathcal{B} \to \mathcal{B}$	$\to$I 2–3

This should not seem so strange, though. Since $\mathcal{B} \to \mathcal{B}$ is a tautology, no particular premises should be required to validly derive it. (Indeed, as we will see, a tautology follows from any premises.)

Put in a general form, the $\to$I rule looks like this:

m	$\quad \mathcal{A}$	want $\mathcal{B}$
n	$\quad \mathcal{B}$	
	$\mathcal{A} \to \mathcal{B}$	$\to$I m–n

When we introduce a subproof, we typically write what we want to derive in the column. This is just so that we do not forget why we started the subproof if it goes on for five or ten lines. There is no 'want' rule. It is a note to ourselves and not formally part of the proof.

Although it is always permissible to open a subproof with any assumption you please, there is some strategy involved in picking a useful assumption. Starting a subproof with an arbitrary, wacky assumption would just waste lines of the proof. In order to derive a conditional by the $\rightarrow$I, for instance, you must assume the antecedent of the conditional in a subproof.

The $\rightarrow$I rule also requires that the consequent of the conditional be the last line of the subproof. It is always permissible to close a subproof and discharge its assumptions, but it will not be helpful to do so until you get what you want.

Now consider the conditional elimination rule. Nothing follows from $M \rightarrow N$ alone, but if we have both $M \rightarrow N$ and M, then we can conclude N. This rule, modus ponens, will be the conditional elimination rule ($\rightarrow$E).

$$
\begin{array}{r|ll}
m & \mathcal{A} \rightarrow \mathcal{B} & \\
n & \mathcal{A} & \\
 & \mathcal{B} & \rightarrow\text{E } m, n
\end{array}
$$

Now that we have rules for the conditional, consider this argument:

$$
\begin{array}{rl}
 & P \rightarrow Q \\
 & Q \rightarrow R \\
\therefore & P \rightarrow R
\end{array}
$$

We begin the proof by writing the two premises as assumptions. Since the main logical operator in the conclusion is a conditional, we can expect to use the $\rightarrow$I rule. For that, we need a subproof— so we write in the antecedent of the conditional as assumption of a subproof:

$$
\begin{array}{r|l}
1 & P \rightarrow Q \\
2 & Q \rightarrow R \\
\hline
3 & \quad \begin{array}{|l} P \\ \hline \end{array}
\end{array}
$$

We made P available by assuming it in a subproof, allowing us to use $\rightarrow$E on the first premise. This gives us Q, which allows us to use $\rightarrow$E on the second premise. Having derived R, we close the subproof. By assuming P we were able to prove R, so we apply the $\rightarrow$I rule and finish the proof.

$$
\begin{array}{r|lll}
1 & P \rightarrow Q & & \\
2 & Q \rightarrow R & & \\
\hline
3 & & \multicolumn{1}{|l}{P} & \text{want } R \\
\cline{3-3}
4 & & \multicolumn{1}{|l}{Q} & \rightarrow\text{E } 1, 3 \\
5 & & \multicolumn{1}{|l}{R} & \rightarrow\text{E } 2, 4 \\
6 & P \rightarrow R & & \rightarrow\text{I } 3\text{–}5
\end{array}
$$

Biconditional

The rules for the biconditional will be like double-barreled versions of the rules for the conditional.

In order to derive $W \leftrightarrow X$, for instance, you must be able to prove X by assuming W *and* prove W by assuming X. The biconditional introduction rule ($\leftrightarrow$I) requires two subproofs. The subproofs can come in any order, and the second subproof does not need to come immediately after the first— but schematically, the rule works like this:

m	$\quad\mathcal{A}$	want $\mathcal{B}$
n	$\quad\mathcal{B}$	
p	$\quad\mathcal{B}$	want $\mathcal{A}$
q	$\quad\mathcal{A}$	
	$\mathcal{A} \leftrightarrow \mathcal{B}$	$\leftrightarrow$I m–n, p–q

The biconditional elimination rule ($\leftrightarrow$E) lets you do a bit more than the conditional rule. If you have the left-hand subsentence of the biconditional, you can derive the right-hand subsentence. If you have the right-hand subsentence, you can derive the left-hand subsentence. This is the rule:

m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{A}$	
	$\mathcal{B}$	$\leftrightarrow$E m, n

m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{B}$	
	$\mathcal{A}$	$\leftrightarrow$E m, n

Negation

Here is a simple mathematical argument in English:

Assume there is some greatest natural number. Call it A.
That number plus one is also a natural number.
Obviously, $A + 1 > A$.
So there is a natural number greater than A.
This is impossible, since A is assumed to be the greatest natural number.
$\therefore$ There is no greatest natural number.

This argument form is traditionally called a *reductio*. Its full Latin name is *reductio ad absurdum*, which means 'reduction to absurdity.' In a reductio, we assume something for the sake of argument— for example, that there is a greatest natural number. Then we show that the assumption leads to two contradictory sentences— for example, that A is the greatest natural number and that it is not. In this way, we show that the original assumption must have been false.

The basic rules for negation will allow for arguments like this. If we assume something and show that it leads to contradictory sentences, then we have proven the negation of the assumption. This is the negation introduction (¬I) rule:

m	$\quad\mathcal{A}$	for reductio
n	$\quad\mathcal{B}$	
$n+1$	$\quad\neg\mathcal{B}$	
$n+2$	$\neg\mathcal{A}$	¬I m–$n+1$

For the rule to apply, the last two lines of the subproof must be an explicit contradiction: some sentence followed on the next line by its negation. We write 'for reductio' as a note to ourselves, a reminder of why we started the subproof. It is not formally part of the proof, and you can leave it out if you find it distracting.

To see how the rule works, suppose we want to prove the law of non-contradiction: $\neg(G \,\&\, \neg G)$. We can prove this without any premises by immediately starting a subproof. We want to apply ¬I to the subproof, so we assume $(G \,\&\, \neg G)$. We then get an explicit contradiction by &E. The proof looks like this:

1	$\quad G \,\&\, \neg G$	for reductio
2	$\quad G$	&E 1
3	$\quad \neg G$	&E 1
4	$\neg(G \,\&\, \neg G)$	¬I 1–3

The ¬E rule will work in much the same way. If we assume $\neg\mathcal{A}$ and show that it leads to a contradiction, we have effectively proven $\mathcal{A}$. So the rule looks like this:

m	$\quad\neg\mathcal{A}$	for reductio
n	$\quad\mathcal{B}$	
$n+1$	$\quad\neg\mathcal{B}$	
$n+2$	$\mathcal{A}$	¬E m–$n+1$

6.2 Derived rules

The rules of the natural deduction system are meant to be systematic. There is an introduction and an elimination rule for each logical operator, but why these basic rules rather than some others? Many natural deduction systems have a disjunction elimination rule that works like this:

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\mathcal{A} \rightarrow \mathcal{C}$	
o	$\mathcal{B} \rightarrow \mathcal{C}$	
	$\mathcal{C}$	DIL m, n, o

Let's call this rule Dilemma (DIL) It might seem as if there will be some proofs that we cannot do with our proof system, because we do not have this as a basic rule. Yet this is not the case. Any proof that you can do using the Dilemma rule can be done with basic rules of our natural deduction system. Consider this proof:

1	$\mathcal{A} \vee \mathcal{B}$	
2	$\mathcal{A} \rightarrow \mathcal{C}$	
3	$\mathcal{B} \rightarrow \mathcal{C}$	want $\mathcal{C}$
4	$\quad \neg\mathcal{C}$	for reductio
5	$\quad\quad \mathcal{A}$	for reductio
6	$\quad\quad \mathcal{C}$	$\rightarrow$E 2, 5
7	$\quad\quad \neg\mathcal{C}$	R 4
8	$\quad \neg\mathcal{A}$	$\neg$I 5–7
9	$\quad\quad \mathcal{B}$	for reductio
10	$\quad\quad \mathcal{C}$	$\rightarrow$E 3, 9
11	$\quad\quad \neg\mathcal{C}$	R 4
12	$\quad \mathcal{B}$	$\vee$E 1, 8
13	$\quad \neg\mathcal{B}$	$\neg$I 9–11
14	$\mathcal{C}$	$\neg$E 4–13

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are meta-variables. They are not symbols of SL, but stand-ins for arbitrary sentences of SL. So this is not, strictly speaking, a proof in SL. It is more like a recipe. It provides a pattern that can prove anything that the Dilemma rule can prove, using only the basic rules of SL. This means that the

Dilemma rule is not really necessary. Adding it to the list of basic rules would not allow us to derive anything that we could not derive without it.

Nevertheless, the Dilemma rule would be convenient. It would allow us to do in one line what requires eleven lines and several nested subproofs with the basic rules. So we will add it to the proof system as a derived rule.

A DERIVED RULE is a rule of proof that does not make any new proofs possible. Anything that can be proven with a derived rule can be proven without it. You can think of a short proof using a derived rule as shorthand for a longer proof that uses only the basic rules. Anytime you use the Dilemma rule, you could always take ten extra lines and prove the same thing without it.

For the sake of convenience, we will add several other derived rules. One is *modus tollens* (MT).

m	$\mathcal{A} \rightarrow \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\neg\mathcal{A}$	MT m, n

We leave the proof of this rule as an exercise. Note that if we had already proven the MT rule, then the proof of the DIL rule could have been done in only five lines.

We also add hypothetical syllogism (HS) as a derived rule. We have already given a proof of it on p. 112.

m	$\mathcal{A} \rightarrow \mathcal{B}$	
n	$\mathcal{B} \rightarrow \mathcal{C}$	
	$\mathcal{A} \rightarrow \mathcal{C}$	HS m, n

6.3 Rules of replacement

Consider how you would prove this argument: $F \rightarrow (G \,\&\, H)$, $\therefore F \rightarrow G$

Perhaps it is tempting to write down the premise and apply the $\&$E rule to the conjunction $(G \,\&\, H)$. This is impermissible, however, because the basic rules of proof can only be applied to whole sentences. We need to get $(G \,\&\, H)$ on a line by itself. We can prove the argument in this way:

1	$F \to (G \,\&\, H)$	
2	$\quad F$	want G
3	$\quad G \,\&\, H$	$\to$E 1, 2
4	$\quad G$	$\&$E 3
5	$F \to G$	$\to$I 2–4

We will now introduce some derived rules that may be applied to part of a sentence. These are called RULES OF REPLACEMENT, because they can be used to replace part of a sentence with a logically equivalent expression. One simple rule of replacement is commutivity (abbreviated Comm), which says that we can swap the order of conjuncts in a conjunction or the order of disjuncts in a disjunction. We define the rule this way:

$$\begin{aligned}(\mathcal{A} \,\&\, \mathcal{B}) &\Longleftrightarrow (\mathcal{B} \,\&\, \mathcal{A})\\ (\mathcal{A} \lor \mathcal{B}) &\Longleftrightarrow (\mathcal{B} \lor \mathcal{A})\\ (\mathcal{A} \leftrightarrow \mathcal{B}) &\Longleftrightarrow (\mathcal{B} \leftrightarrow \mathcal{A}) \quad \text{Comm}\end{aligned}$$

The bold arrow means that you can take a subformula on one side of the arrow and replace it with the subformula on the other side. The arrow is double-headed because rules of replacement work in both directions.

Consider this argument: $(M \lor P) \to (P \,\&\, M)$, $\therefore$ $(P \lor M) \to (M \,\&\, P)$

It is possible to give a proof of this using only the basic rules, but it will be long and inconvenient. With the Comm rule, we can provide a proof easily:

1	$(M \lor P) \to (P \,\&\, M)$	
2	$(P \lor M) \to (P \,\&\, M)$	Comm 1
3	$(P \lor M) \to (M \,\&\, P)$	Comm 2

Another rule of replacement is double negation (DN). With the DN rule, you can remove or insert a pair of negations anywhere in a sentence. This is the rule:

$$\neg\neg\mathcal{A} \Longleftrightarrow \mathcal{A} \quad \text{DN}$$

Two more replacement rules are called De Morgan's Laws, named for the 19th-century British logician August De Morgan. (Although De Morgan did discover these laws, he was not the first to do so.) The rules capture useful relations between negation, conjunction, and disjunction. Here are the rules, which we abbreviate DeM:

$$\neg(\mathcal{A} \vee \mathcal{B}) \iff (\neg\mathcal{A} \,\&\, \neg\mathcal{B})$$
$$\neg(\mathcal{A} \,\&\, \mathcal{B}) \iff (\neg\mathcal{A} \vee \neg\mathcal{B}) \quad \text{DeM}$$

Because $\mathcal{A} \rightarrow \mathcal{B}$ is a *material conditional*, it is equivalent to $\neg\mathcal{A} \vee \mathcal{B}$. A further replacement rule captures this equivalence. We abbreviate the rule MC, for 'material conditional.' It takes two forms:

$$(\mathcal{A} \rightarrow \mathcal{B}) \iff (\neg\mathcal{A} \vee \mathcal{B})$$
$$(\mathcal{A} \vee \mathcal{B}) \iff (\neg\mathcal{A} \rightarrow \mathcal{B}) \quad \text{MC}$$

Now consider this argument: $\neg(P \rightarrow Q)$, $\therefore$ $P \,\&\, \neg Q$

As always, we could prove this argument using only the basic rules. With rules of replacement, though, the proof is much simpler:

1	$\neg(P \rightarrow Q)$	
2	$\neg(\neg P \vee Q)$	MC 1
3	$\neg\neg P \,\&\, \neg Q$	DeM 2
4	$P \,\&\, \neg Q$	DN 3

A final replacement rule captures the relation between conditionals and biconditionals. We will call this rule biconditional exchange and abbreviate it $\leftrightarrow$ex.

$$[(\mathcal{A} \rightarrow \mathcal{B}) \,\&\, (\mathcal{B} \rightarrow \mathcal{A})] \iff (\mathcal{A} \leftrightarrow \mathcal{B}) \quad \leftrightarrow\text{ex}$$

6.4 Rules for quantifiers

For proofs in QL, we use all of the basic rules of SL plus four new basic rules: both introduction and elimination rules for each of the quantifiers.

Since all of the derived rules of SL are derived from the basic rules, they will also hold in QL. We will add another derived rule, a replacement rule called quantifier negation.

Substitution instances

In order to concisely state the rules for the quantifiers, we need a way to mark the relation between quantified sentences and their instances. For example, the sentence Pa is a particular instance of the general claim $\forall x Px$.

For a wff $\mathscr{A}$, a constant $\mathscr{c}$, and a variable $\mathscr{x}$, define a SUBSTITUTION INSTANCE of $\forall\mathscr{x}\mathscr{A}$ or $\exists\mathscr{x}\mathscr{A}$ is the wff that we get by replacing every occurrence of $\mathscr{x}$ in $\mathscr{A}$ with $\mathscr{c}$. We call $\mathscr{c}$ the INSTANTIATING CONSTANT.

To underscore the fact that the variable $\mathscr{x}$ is replaced by the instantiating constant $\mathscr{c}$, we will write the original quantified expressions as $\forall\mathscr{x}\mathscr{A}\mathscr{x}$ and $\exists\mathscr{x}\mathscr{A}\mathscr{x}$. And we will write the substitution instance $\mathscr{A}\mathscr{c}$.

Note that $\mathscr{A}$, $\mathscr{x}$, and $\mathscr{c}$ are all meta-variables. That is, they are stand-ins for any wff, variable, and constant whatsoever. And when we write $\mathscr{A}\mathscr{c}$, the constant $\mathscr{c}$ may occur multiple times in the wff $\mathscr{A}$.

For example:

- $Aa \to Ba$, $Af \to Bf$, and $Ak \to Bk$ are all substitution instances of $\forall x(Ax \to Bx)$; the instantiating constants are a, f, and k, respectively.
- Raj, Rdj, and Rjj are substitution instances of $\exists zRzj$; the instantiating constants are a, d, and j, respectively.

Universal elimination

If you have $\forall xAx$, it is legitimate to infer that anything is an A. You can infer Aa, Ab, Az, Ad_3. You can infer any substitution instance, $A\mathscr{c}$ for any constant $\mathscr{c}$.

This is the general form of the universal elimination rule ($\forall$E):

m	$\forall\mathscr{x}\mathscr{A}\mathscr{x}$	
	$\mathscr{A}\mathscr{c}$	$\forall$E m

When using the $\forall$E rule, you write the substituted sentence with the constant $\mathscr{c}$ replacing all occurrences of the variable $\mathscr{x}$ in $\mathscr{A}$. For example:

1	$\forall x(Mx \to Rxd)$	
2	$Ma \to Rad$	$\forall$E 1
3	$Md \to Rdd$	$\forall$E 1

Existential introduction

It is legitimate to infer $\exists xPx$ if you know that *something* is a P. It might be any particular thing at all. For example, if you have Pa available in the proof, then $\exists xPx$ follows.

This is the existential introduction rule (∃I):

m	$\mathcal{A}c$	
	$\exists\chi\mathcal{A}\chi$	∃I m

It is important to notice that the variable χ does not need to replace all occurrences of the constant c. You can decide which occurrences to replace and which to leave in place. For example:

1	$Ma \to Rad$	
2	$\exists x(Ma \to Rax)$	∃I 1
3	$\exists x(Mx \to Rxd)$	∃I 1
4	$\exists x(Mx \to Rad)$	∃I 1
5	$\exists y\exists x(Mx \to Ryd)$	∃I 4
6	$\exists z\exists y\exists x(Mx \to Ryz)$	∃I 5

Universal introduction

A universal claim like $\forall xPx$ would be proven if every substitution instance of it had been proven. That is, if every sentence Pa, Pb, ... were available in a proof, then you would certainly be entitled to claim $\forall xPx$. Alas, there is no hope of proving *every* substitution instance. That would require proving Pa, Pb, ..., Pj_2, ..., Ps_7, ..., and so on to infinity. There are infinitely many constants in QL, and so this process would never come to an end.

Consider instead a simple argument: $\forall xMx$, $\therefore$ $\forall yMy$

It makes no difference to the meaning of the sentence whether we use the variable x or the variable y, so this argument is obviously valid. Suppose we begin in this way:

1	$\forall xMx$	want $\forall yMy$
2	Ma	∀E 1

We have derived Ma. Nothing stops us from using the same justification to derive Mb, ..., Mj_2, ..., Ms_7, ..., and so on until we run out of space or patience. We have effectively shown the way to prove Mc for any constant c. From this, $\forall yMy$ follows.

1	$\forall x M x$	
2	Ma	$\forall$E 1
3	$\forall y M y$	$\forall$I 2

It is important here that a was just some arbitrary constant. We had not made any special assumptions about it. If Ma were a premise of the argument, then this would not show anything about *all* y. For example:

1	$\forall x Rxa$	
2	Raa	$\forall$E 1
3	$\forall y Ryy$	not allowed!

This is the schematic form of the universal introduction rule ($\forall$I):

m	$\mathcal{A}c^*$	
	$\forall \chi \mathcal{A}\chi$	$\forall$I m

* The constant c must not occur in any undischarged assumption.

Note that we can do this for any constant that does not occur in an undischarged assumption and for any variable.

Note also that the constant may not occur in any *undischarged* assumption, but it may occur as the assumption of a subproof that we have already closed. For example, we can prove $\forall z(Dz \rightarrow Dz)$ without any premises.

1	$\quad Df$	want Df
2	$\quad Df$	R 1
3	$Df \rightarrow Df$	$\rightarrow$I 1–2
4	$\forall z(Dz \rightarrow Dz)$	$\forall$I 3

Existential elimination

A sentence with an existential quantifier tells us that there is *some* member of the UD that satisfies a formula. For example, $\exists x Sx$ tells us (roughly) that there is at least one S. It does not tell us *which* member of the UD satisfies S, however. We cannot immediately conclude Sa, Sf_{23}, or any other substitution instance of the sentence. What can we do?

Suppose that we knew both $\exists x Sx$ and $\forall x(Sx \rightarrow Tx)$. We could reason in this way:

> Since $\exists x Sx$, there is something that is an S. We do not know which constants refer to this thing, if any do, so call this thing 'Ishmael'. From $\forall x(Sx \rightarrow Tx)$, it follows that if Ishmael is an S, then it is a T. Therefore, Ishmael is a T. Because Ishmael is a T, we know that $\exists x Tx$.

In this paragraph, we introduced a name for the thing that is an S. We gave it an arbitrary name ('Ishmael') so that we could reason about it and derive some consequences from there being an S. Since 'Ishmael' is just a bogus name introduced for the purpose of the proof and not a genuine constant, we could not mention it in the conclusion. Yet we could derive a sentence that does not mention Ishmael; namely, $\exists x Tx$. This sentence does follow from the two premises.

We want the existential elimination rule to work in a similar way. Yet since English language words like 'Ishmael' are not symbols of QL, we cannot use them in formal proofs. Instead, we will use constants of QL which do not otherwise appear in the proof.

A constant that is used to stand in for whatever it is that satisfies an existential claim is called a PROXY. Reasoning with the proxy must all occur inside a subproof, and the proxy cannot be a constant that is doing work elsewhere in the proof.

This is the schematic form of the existential elimination rule ($\exists$E):

$$\begin{array}{l|l l}
m & \exists \chi \mathcal{A}\chi & \\
n & \quad \mathcal{A}c^{*} & \\
p & \quad \mathcal{B} & \\
 & \mathcal{B} & \exists\text{E } m,\ n\text{–}p
\end{array}$$

* The constant c must not appear in $\exists \chi \mathcal{A}\chi$, in $\mathcal{B}$, or in any undischarged assumption.

Since the proxy constant is just a place holder that we use inside the subproof, it cannot be something that we know anything particular about. So it cannot appear in the original sentence $\exists \chi \mathcal{A}\chi$ or in an undischarged assumption. Moreover, we do not learn anything about the proxy constant by using the $\exists$E rule. So it cannot appear in $\mathcal{B}$, the sentence you prove using $\exists$E.

The easiest way to satisfy these requirements is to pick an entirely new constant when you start the subproof, and then not to use that constant anywhere else in the proof. Once you close the subproof, do not mention it again.

With this rule, we can give a formal proof that $\exists x Sx$ and $\forall x(Sx \rightarrow Tx)$ together entail $\exists x Tx$.

1	$\exists x Sx$	
2	$\forall x(Sx \to Tx)$	want $\exists x Tx$
3	$\quad Si$	
4	$\quad Si \to Ti$	$\forall$E 2
5	$\quad Ti$	$\to$E 3, 4
6	$\quad \exists x Tx$	$\exists$I 5
7	$\exists x Tx$	$\exists$E 1, 3–6

Notice that this has effectively the same structure as the English-language argument with which we began, except that the subproof uses the proxy constant 'i' rather than the bogus name 'Ishmael'.

Quantifier negation

When translating from English to QL, we noted that $\neg\exists x\neg\mathcal{A}$ is logically equivalent to $\forall x\mathcal{A}$. In QL, they are provably equivalent. We can prove one half of the equivalence with a rather gruesome proof:

1	$\forall x Ax$	want $\neg\exists x\neg Ax$
2	$\quad \exists x\neg Ax$	for reductio
3	$\quad\quad \neg Ac$	for $\exists$E
4	$\quad\quad\quad \forall x Ax$	for reductio
5	$\quad\quad\quad Ac$	$\forall$E 1
6	$\quad\quad\quad \neg Ac$	R 3
7	$\quad\quad \neg\forall x Ax$	$\neg$I 4–6
8	$\quad \forall x Ax$	R 1
9	$\quad \neg\forall x Ax$	$\exists$E 2, 3–7
10	$\neg\exists x\neg Ax$	$\neg$I 2–9

In order to show that the two sentences are genuinely equivalent, we need a second proof that assumes $\neg\exists x\neg\mathcal{A}$ and derives $\forall x\mathcal{A}$. We leave that proof as an exercise for the reader.

It will often be useful to translate between quantifiers by adding or subtracting negations in this way, so we add two derived rules for this purpose. These rules are called quantifier negation (QN):

$$\neg\forall\mathcal{X}\mathcal{A} \Longleftrightarrow \exists\mathcal{X}\neg\mathcal{A}$$
$$\neg\exists\mathcal{X}\mathcal{A} \Longleftrightarrow \forall\mathcal{X}\neg\mathcal{A} \quad \text{QN}$$

Since QN is a replacement rule, it can be used on whole sentences or on subformulae.

6.5 Rules for identity

The identity predicate is not part of QL, but we add it when we need to symbolize certain sentences. For proofs involving identity, we add two rules of proof.

Suppose you know that many things that are true of a are also true of b. For example: $Aa \& Ab$, $Ba \& Bb$, $\neg Ca \& \neg Cb$, $Da \& Db$, $\neg Ea \& \neg Eb$, and so on. This would not be enough to justify the conclusion $a = b$. (See p. 91.) In general, there are no sentences that do not already contain the identity predicate that could justify the conclusion $a = b$. This means that the identity introduction rule will not justify $a = b$ or any other identity claim containing two different constants.

However, it is always true that $a = a$. In general, no premises are required in order to conclude that something is identical to itself. So this will be the identity introduction rule, abbreviated =I:

| $\mathcal{c} = \mathcal{c}$ =I

Notice that the =I rule does not require referring to any prior lines of the proof. For any constant $\mathcal{c}$, you can write $\mathcal{c} = \mathcal{c}$ on any point with only the =I rule as justification.

If you have shown that $a = b$, then anything that is true of a must also be true of b. For any sentence with a in it, you can replace some or all of the occurrences of a with b and produce an equivalent sentence. For example, if you already know Raa, then you are justified in concluding Rab, Rba, Rbb.

The identity elimination rule (=E) allows us to do this. It justifies replacing terms with other terms that are identical to it.

For writing the rule, we will introduce a new bit of symbolism. For a sentence $\mathcal{A}$ and constants $\mathcal{c}$ and $\mathcal{d}$, $\mathcal{A}\mathcal{c}\circlearrowleft\mathcal{d}$ is a sentence produced by replacing some or all instances of $\mathcal{c}$ in $\mathcal{A}$ with $\mathcal{d}$ or replacing instances of $\mathcal{d}$ with $\mathcal{c}$. This is not the same as a substitution instance, because one constant need not replace every occurrence of the other (although it may).

We can now concisely write =E in this way:

m	$\mathcal{c} = \mathcal{d}$	
n	$\mathcal{A}$	
	$\mathcal{A}\mathcal{c} \circlearrowleft \mathcal{d}$	$=$E m, n

To see the rules in action, consider this proof:

1	$\forall x \forall y\ x = y$	
2	$\exists x Bx$	
3	$\forall x(Bx \to \neg Cx)$	want $\neg\exists x Cx$
4	$\quad$ Be	
5	$\quad$ $\forall y\ e = y$	$\forall$E 1
6	$\quad$ $e = f$	$\forall$E 5
7	$\quad$ Bf	$=$E 6, 4
8	$\quad$ $Bf \to \neg Cf$	$\forall$E 3
9	$\quad$ $\neg Cf$	$\to$E 8, 7
10	$\neg Cf$	$\exists$E 2, 4–9
11	$\forall x \neg Cx$	$\forall$I 10
12	$\neg\exists x Cx$	QN 11

6.6 Proof strategy

There is no simple recipe for proofs, and there is no substitute for practice. Here, though, are some rules of thumb and strategies to keep in mind.

Work backwards from what you want. The ultimate goal is to derive the conclusion. Look at the conclusion and ask what the introduction rule is for its main logical operator. This gives you an idea of what should happen *just before* the last line of the proof. Then you can treat this line as if it were your goal. Ask what you could do to derive this new goal.

For example: If your conclusion is a conditional $\mathcal{A} \to \mathcal{B}$, plan to use the $\to$I rule. This requires starting a subproof in which you assume $\mathcal{A}$. In the subproof, you want to derive $\mathcal{B}$.

Work forwards from what you have. When you are starting a proof, look at the premises; later, look at the sentences that you have derived so far. Think about the elimination rules for the main operators of these sentences. These will tell you what your options are.

For example: If you have $\forall x \mathcal{A}$, think about instantiating it for any constant that might be helpful. If you have $\exists x \mathcal{A}$ and intend to use the $\exists$E rule, then you should assume $\mathcal{A}[c|x]$ for some c that is not in use and then derive a conclusion that does not contain c.

For a short proof, you might be able to eliminate the premises and introduce the conclusion. A long proof is formally just a number of short proofs linked together, so you can fill the gap by alternately working back from the conclusion and forward from the premises.

Change what you are looking at. Replacement rules can often make your life easier. If a proof seems impossible, try out some different substitutions.

For example: It is often difficult to prove a disjunction using the basic rules. If you want to show $\mathcal{A} \vee \mathcal{B}$, it is often easier to show $\neg\mathcal{A} \rightarrow \mathcal{B}$ and use the MC rule.

Showing $\neg\exists x \mathcal{A}$ can also be hard, and it is often easier to show $\forall x \neg\mathcal{A}$ and use the QN rule.

Some replacement rules should become second nature. If you see a negated disjunction, for instance, you should immediately think of DeMorgan's rule.

Do not forget indirect proof. If you cannot find a way to show something directly, try assuming its negation.

Remember that most proofs can be done either indirectly or directly. One way might be easier— or perhaps one sparks your imagination more than the other— but either one is formally legitimate.

Repeat as necessary. Once you have decided how you might be able to get to the conclusion, ask what you might be able to do with the premises. Then consider the target sentences again and ask how you might reach them.

Persist. Try different things. If one approach fails, then try something else.

6.7 Proof-theoretic concepts

We will use the symbol '$\vdash$' to indicate that a proof is possible. This symbol is called the *turnstile*. Sometimes it is called a *single turnstile*, to underscore the fact that this is not the double turnstile symbol ($\models$) that we used to represent semantic entailment in ch. 5.

When we write $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash \mathcal{B}$, this means that it is possible to give a proof of $\mathcal{B}$ with $\mathcal{A}_1$,$\mathcal{A}_2$,... as premises. With just one premise, we leave out the curly braces, so $\mathcal{A} \vdash \mathcal{B}$ means that there is a proof of $\mathcal{B}$ with $\mathcal{A}$ as a premise. Naturally, $\vdash \mathcal{C}$ means that there is a proof of $\mathcal{C}$ that has no premises.

Often, logical proofs are called *derivations*. So $\mathcal{A} \vdash \mathcal{B}$ can be read as '$\mathcal{B}$ is derivable from $\mathcal{A}$.'

A THEOREM is a sentence that is derivable without any premises; i.e., $\mathcal{T}$ is a theorem if and only if $\vdash \mathcal{T}$.

It is not too hard to show that something is a theorem— you just have to give a proof of it. How could you show that something is *not* a theorem? If its negation is a theorem, then you could provide a proof. For example, it is easy to prove $\neg(Pa \,\&\, \neg Pa)$, which shows that $(Pa \,\&\, \neg Pa)$ cannot be a theorem. For a sentence that is neither a theorem nor the negation of a theorem, however, there is no easy way to show this. You would have to demonstrate not just that certain proof strategies fail, but that no proof is possible. Even if you fail in trying to prove a sentence in a thousand different ways, perhaps the proof is just too long and complex for you to make out.

Two sentences $\mathcal{A}$ and $\mathcal{B}$ are PROVABLY EQUIVALENT if and only if each can be derived from the other; i.e., $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$

It is relatively easy to show that two sentences are provably equivalent— it just requires a pair of proofs. Showing that sentences are *not* provably equivalent would be much harder. It would be just as hard as showing that a sentence is not a theorem. (In fact, these problems are interchangeable. Can you think of a sentence that would be a theorem if and only if $\mathcal{A}$ and $\mathcal{B}$ were provably equivalent?)

The set of sentences $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\}$ is PROVABLY INCONSISTENT if and only if a contradiction is derivable from it; i.e., for some sentence $\mathcal{B}$, $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash \mathcal{B}$ and $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash \neg\mathcal{B}$.

It is easy to show that a set is provably inconsistent: You just need to assume the sentences in the set and prove a contradiction. Showing that a set is *not* provably inconsistent will be much harder. It would require more than just providing a proof or two; it would require showing that proofs of a certain kind are *impossible*.

6.8 Proofs and models

As you might already suspect, there is a connection between *theorems* and *tautologies*.

There is a formal way of showing that a sentence is a theorem: Prove it. For each line, we can check to see if that line follows by the cited rule. It may be hard to produce a twenty line proof, but it is not so hard to check each line of the proof and confirm that it is legitimate— and if each line of the proof individually is legitimate, then the whole proof is legitimate. Showing that a sentence is a tautology, though, requires reasoning in English about all possible models. There is no formal way of checking to see if the reasoning is sound. Given a choice between showing that a sentence is a theorem and showing that it is a tautology, it would be easier to show that it is a theorem.

Contrawise, there is no formal way of showing that a sentence is *not* a theorem. We would need to reason in English about all possible proofs. Yet there is a formal method for showing that a sentence is not a tautology. We need only construct a model in which the sentence is false. Given a choice between showing that a sentence is not a theorem and showing that it is not a tautology, it would be easier to show that it is not a tautology.

Fortunately, a sentence is a theorem if and only if it is a tautology. If we provide a proof of $\vdash \mathcal{A}$ and thus show that it is a theorem, it follows that $\mathcal{A}$ is a tautology; i.e., $\models \mathcal{A}$. Similarly, if we construct a model in which $\mathcal{A}$ is false and thus show that it is not a tautology, it follows that $\mathcal{A}$ is not a theorem.

In general, $\mathcal{A} \vdash \mathcal{B}$ if and only if $\mathcal{A} \models \mathcal{B}$. As such:

- An argument is *valid* if and only if *the conclusion is derivable from the premises.*
- Two sentences are *logically equivalent* if and only if they are *provably equivalent.*
- A set of sentences is *consistent* if and only if it is *not provably inconsistent.*

You can pick and choose when to think in terms of proofs and when to think in terms of models, doing whichever is easier for a given task. Table 6.1 summarizes when it is best to give proofs and when it is best to give models.

In this way, proofs and models give us a versatile toolkit for working with arguments. If we can translate an argument into QL, then we can measure its logical weight in a purely formal way. If it is deductively valid, we can give a formal proof; if it is invalid, we can provide a formal counterexample.

	YES	NO
Is $\mathcal{A}$ a tautology?	prove $\vdash \mathcal{A}$	give a model in which $\mathcal{A}$ is false
Is $\mathcal{A}$ a contradiction?	prove $\vdash \neg\mathcal{A}$	give a model in which $\mathcal{A}$ is true
Is $\mathcal{A}$ contingent?	give a model in which $\mathcal{A}$ is true and another in which $\mathcal{A}$ is false	prove $\vdash \mathcal{A}$ or $\vdash \neg\mathcal{A}$
Are $\mathcal{A}$ and $\mathcal{B}$ equivalent?	prove $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$	give a model in which $\mathcal{A}$ and $\mathcal{B}$ have different truth values
Is the set $\mathbb{A}$ consistent?	give a model in which all the sentences in $\mathbb{A}$ are true	taking the sentences in $\mathbb{A}$, prove $\mathcal{B}$ and $\neg\mathcal{B}$
Is the argument '$\mathcal{P}$, $\therefore$ $\mathcal{C}$' valid?	prove $\mathcal{P} \vdash \mathcal{C}$	give a model in which $\mathcal{P}$ is true and $\mathcal{C}$ is false

Tabela 6.1: Sometimes it is easier to show something by providing proofs than it is by providing models. Sometimes it is the other way round. It depends on what you are trying to show.

6.9 Soundness and completeness

This toolkit is incredibly convenient. It is also intuitive, because it seems natural that provability and semantic entailment should agree. Yet, do not be fooled by the similarity of the symbols '$\models$' and '$\vdash$.' The fact that these two are really interchangeable is not a simple thing to prove.

Why should we think that an argument that *can be proven* is necessarily a *valid* argument? That is, why think that $\mathcal{A} \vdash \mathcal{B}$ implies $\mathcal{A} \models \mathcal{B}$?

This is the problem of SOUNDNESS. A proof system is SOUND if there are no proofs of invalid arguments. Demonstrating that the proof system is sound would require showing that *any* possible proof is the proof of a valid argument. It would not be enough simply to succeed when trying to prove many valid arguments and to fail when trying to prove invalid ones.

Fortunately, there is a way of approaching this in a step-wise fashion. If using the $\&$E rule on the last line of a proof could never change a valid argument into an invalid one, then using the rule many times could not make an argument invalid. Similarly, if using the $\&$E and $\vee$E rules individually on the last line of a proof could never change a valid argument into an invalid one, then using them in combination could not either.

The strategy is to show for every rule of inference that it alone could not make a valid argument into an invalid one. It follows that the rules used in combination would not make a valid argument invalid. Since a proof is just a series of lines, each justified by a rule of inference, this would show that every provable argument is valid.

Consider, for example, the &I rule. Suppose we use it to add $\mathcal{A} \& \mathcal{B}$ to a valid argument. In order for the rule to apply, $\mathcal{A}$ and $\mathcal{B}$ must already be available in the proof. Since the argument so far is valid, $\mathcal{A}$ and $\mathcal{B}$ are either premises of the argument or valid consequences of the premises. As such, any model in which the premises are true must be a model in which $\mathcal{A}$ and $\mathcal{B}$ are true. According to the definition of TRUTH IN QL, this means that $\mathcal{A} \& \mathcal{B}$ is also true in such a model. Therefore, $\mathcal{A} \& \mathcal{B}$ validly follows from the premises. This means that using the &E rule to extend a valid proof produces another valid proof.

In order to show that the proof system is sound, we would need to show this for the other inference rules. Since the derived rules are consequences of the basic rules, it would suffice to provide similar arguments for the 16 other basic rules. This tedious exercise falls beyond the scope of this book.

Given a proof that the proof system is sound, it follows that every theorem is a tautology.

It is still possible to ask: Why think that *every* valid argument is an argument that can be proven? That is, why think that $\mathcal{A} \models \mathcal{B}$ implies $\mathcal{A} \vdash \mathcal{B}$?

This is the problem of COMPLETENESS. A proof system is COMPLETE if there is a proof of every valid argument. Completeness for a language like QL was first proven by Kurt Gödel in 1929. The proof is beyond the scope of this book.

The important point is that, happily, the proof system for QL is both sound and complete. This is not the case for all proof systems and all formal languages. Because it is true of QL, we can choose to give proofs or construct models—whichever is easier for the task at hand.

Summary of definitions

- ▷ A sentence $\mathcal{A}$ is a THEOREM if and only if $\vdash \mathcal{A}$.

- ▷ Two sentences $\mathcal{A}$ and $\mathcal{B}$ are PROVABLY EQUIVALENT if and only if $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$.

- ▷ $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\}$ is PROVABLY INCONSISTENT if and only if, for some sentence $\mathcal{B}$, $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash (\mathcal{B} \& \neg\mathcal{B})$.

Practice Exercises

⋆ **Part A** Provide a justification (rule and line numbers) for each line of proof that requires one.

1	$W \to \neg B$
2	$A \,\&\, W$
3	$B \vee (J \,\&\, K)$
4	W
5	$\neg B$
6	$J \,\&\, K$
7	K

1	$L \leftrightarrow \neg O$
2	$L \vee \neg O$
3	$\quad \neg L$
4	$\quad \neg O$
5	$\quad L$
6	$\quad \neg L$
7	L

1	$Z \to (C \,\&\, \neg N)$
2	$\neg Z \to (N \,\&\, \neg C)$
3	$\quad \neg(N \vee C)$
4	$\quad \neg N \,\&\, \neg C$
5	$\quad\quad Z$
6	$\quad\quad C \,\&\, \neg N$
7	$\quad\quad C$
8	$\quad\quad \neg C$
9	$\quad \neg Z$
10	$\quad N \,\&\, \neg C$
11	$\quad N$
12	$\quad \neg N$
13	$N \vee C$

⋆ **Part B** Give a proof for each argument in SL.

1. $K \,\&\, L, \therefore K \leftrightarrow L$
2. $A \to (B \to C), \therefore (A \,\&\, B) \to C$
3. $P \,\&\, (Q \vee R),\ P \to \neg R, \therefore Q \vee E$
4. $(C \,\&\, D) \vee E, \therefore E \vee D$
5. $\neg F \to G,\ F \to H, \therefore G \vee H$
6. $(X \,\&\, Y) \vee (X \,\&\, Z),\ \neg(X \,\&\, D),\ D \vee M\ \therefore M$

Part C Give a proof for each argument in SL.

1. $Q \to (Q \,\&\, \neg Q), \therefore \neg Q$
2. $J \to \neg J, \therefore \neg J$
3. $E \vee F,\ F \vee G,\ \neg F, \therefore E \,\&\, G$
4. $A \leftrightarrow B,\ B \leftrightarrow C, \therefore A \leftrightarrow C$
5. $M \vee (N \to M), \therefore \neg M \to \neg N$

6. $S \leftrightarrow T$, $\therefore S \leftrightarrow (T \vee S)$
7. $(M \vee N) \,\&\, (O \vee P)$, $N \rightarrow P$, $\neg P$, $\therefore M \,\&\, O$
8. $(Z \,\&\, K) \vee (K \,\&\, M)$, $K \rightarrow D$, $\therefore D$

Part D Show that each of the following sentences is a theorem in SL.

1. $O \rightarrow O$
2. $N \vee \neg N$
3. $\neg(P \,\&\, \neg P)$
4. $\neg(A \rightarrow \neg C) \rightarrow (A \rightarrow C)$
5. $J \leftrightarrow [J \vee (L \,\&\, \neg L)]$

Part E Show that each of the following pairs of sentences are provably equivalent in SL.

1. $\neg\neg\neg\neg G$, G
2. $T \rightarrow S$, $\neg S \rightarrow \neg T$
3. $R \leftrightarrow E$, $E \leftrightarrow R$
4. $\neg G \leftrightarrow H$, $\neg(G \leftrightarrow H)$
5. $U \rightarrow I$, $\neg(U \,\&\, \neg I)$

Part F Provide proofs to show each of the following.

1. $M \,\&\, (\neg N \rightarrow \neg M) \vdash (N \,\&\, M) \vee \neg M$
2. $\{C \rightarrow (E \,\&\, G), \neg C \rightarrow G\} \vdash G$
3. $\{(Z \,\&\, K) \leftrightarrow (Y \,\&\, M), D \,\&\, (D \rightarrow M)\} \vdash Y \rightarrow Z$
4. $\{(W \vee X) \vee (Y \vee Z), X \rightarrow Y, \neg Z\} \vdash W \vee Y$

Part G For the following, provide proofs using only the basic rules. The proofs will be longer than proofs of the same claims would be using the derived rules.

1. Show that MT is a legitimate derived rule. Using only the basic rules, prove the following: $\mathcal{A} \rightarrow \mathcal{B}$, $\neg\mathcal{B}$, $\therefore \neg\mathcal{A}$
2. Show that Comm is a legitimate rule for the biconditional. Using only the basic rules, prove that $\mathcal{A} \leftrightarrow \mathcal{B}$ and $\mathcal{B} \leftrightarrow \mathcal{A}$ are equivalent.
3. Using only the basic rules, prove the following instance of DeMorgan's Laws: $(\neg A \,\&\, \neg B)$, $\therefore \neg(A \vee B)$
4. Without using the QN rule, prove $\neg\exists x \neg\mathcal{A} \vdash \forall x \mathcal{A}$
5. Show that $\leftrightarrow$ex is a legitimate derived rule. Using only the basic rules, prove that $D \leftrightarrow E$ and $(D \rightarrow E) \,\&\, (E \rightarrow D)$ are equivalent.

$\star$ **Part H**

1. Identify which of the following are substitution instances of $\forall x Rcx$: Rac, Rca, Raa, Rcb, Rbc, Rcc, Rcd, Rcx
2. Identify which of the following are substitution instances of $\exists x \forall y Lxy$: $\forall y Lby$, $\forall x Lbx$, Lab, $\exists x Lxa$

⋆ **Part I** Provide a justification (rule and line numbers) for each line of proof that requires one.

$$\begin{array}{r|l}
1 & \forall x\exists y(Rxy \lor Ryx) \\
2 & \underline{\forall x\neg Rmx} \\
3 & \exists y(Rmy \lor Rym) \\
4 & \quad \mid \underline{Rma \lor Ram} \\
5 & \quad \mid \neg Rma \\
6 & \quad \mid Ram \\
7 & \quad \mid \exists x Rxm \\
8 & \exists x Rxm
\end{array}$$

$$\begin{array}{r|l}
1 & \forall x(Jx \to Kx) \\
2 & \exists x\forall y Lxy \\
3 & \underline{\forall x Jx} \\
4 & \quad \mid \underline{\forall y Lay} \\
5 & \quad \mid Ja \\
6 & \quad \mid Ja \to Ka \\
7 & \quad \mid Ka \\
8 & \quad \mid Laa \\
9 & \quad \mid Ka \,\&\, Laa \\
10 & \quad \mid \exists x(Kx \,\&\, Lxx) \\
11 & \exists x(Kx \,\&\, Lxx)
\end{array}$$

$$\begin{array}{r|l}
1 & \forall x(\exists y Lxy \to \forall z Lzx) \\
2 & \underline{Lab} \\
3 & \exists y Lay \to \forall z Lza \\
4 & \exists y Lay \\
5 & \forall z Lza \\
6 & Lca \\
7 & \exists y Lcy \to \forall z Lzc \\
8 & \exists y Lcy \\
9 & \forall z Lzc \\
10 & Lcc \\
11 & \forall x Lxx
\end{array}$$

$$\begin{array}{r|l}
1 & \quad \mid \underline{\neg(\exists x Mx \lor \forall x\neg Mx)} \\
2 & \quad \mid \neg\exists x Mx \,\&\, \neg\forall x\neg Mx \\
3 & \quad \mid \neg\exists x Mx \\
4 & \quad \mid \forall x\neg Mx \\
5 & \quad \mid \neg\forall x\neg Mx \\
6 & \exists x Mx \lor \forall x\neg Mx
\end{array}$$

⋆ **Part J** Provide a proof of each claim.

1. $\vdash \forall x Fx \lor \neg\forall x Fx$
2. $\{\forall x(Mx \leftrightarrow Nx), Ma \,\&\, \exists x Rxa\} \vdash \exists x Nx$
3. $\{\forall x(\neg Mx \lor Ljx), \forall x(Bx \to Ljx), \forall x(Mx \lor Bx)\} \vdash \forall x Ljx$
4. $\forall x(Cx \,\&\, Dt) \vdash \forall x Cx \,\&\, Dt$
5. $\exists x(Cx \lor Dt) \vdash \exists x Cx \lor Dt$

Part K Provide a proof of the argument about Billy on p. 64.

Part L Look back at Part B on p. 75. Provide proofs to show that each of the argument forms is valid in QL.

Part M Aristotle and his successors identified other syllogistic forms. Symbolize each of the following argument forms in QL and add the additional assumptions 'There is an A' and 'There is a B.' Then prove that the supplemented arguments forms are valid in QL.

Darapti: All As are Bs. All As are Cs. $\therefore$ Some B is C.

Felapton: No Bs are Cs. All As are Bs. $\therefore$ Some A is not C.

Barbari: All Bs are Cs. All As are Bs. $\therefore$ Some A is C.

Camestros: All Cs are Bs. No As are Bs. $\therefore$ Some A is not C.

Celaront: No Bs are Cs. All As are Bs. $\therefore$ Some A is not C.

Cesaro: No Cs are Bs. All As are Bs. $\therefore$ Some A is not C.

Fapesmo: All Bs are Cs. No As are Bs. $\therefore$ Some C is not A.

Part N Provide a proof of each claim.

1. $\forall x \forall y Gxy \vdash \exists x Gxx$
2. $\forall x \forall y (Gxy \rightarrow Gyx) \vdash \forall x \forall y (Gxy \leftrightarrow Gyx)$
3. $\{\forall x(Ax \rightarrow Bx), \exists x Ax\} \vdash \exists x Bx$
4. $\{Na \rightarrow \forall x(Mx \leftrightarrow Ma), Ma, \neg Mb\} \vdash \neg Na$
5. $\vdash \forall z(Pz \vee \neg Pz)$
6. $\vdash \forall x Rxx \rightarrow \exists x \exists y Rxy$
7. $\vdash \forall y \exists x(Qy \rightarrow Qx)$

Part O Show that each pair of sentences is provably equivalent.

1. $\forall x(Ax \rightarrow \neg Bx)$, $\neg \exists x(Ax \,\&\, Bx)$
2. $\forall x(\neg Ax \rightarrow Bd)$, $\forall x Ax \vee Bd$
3. $\exists x Px \rightarrow Qc$, $\forall x(Px \rightarrow Qc)$

Part P Show that each of the following is provably inconsistent.

1. $\{Sa \rightarrow Tm,\ Tm \rightarrow Sa,\ Tm \,\&\, \neg Sa\}$
2. $\{\neg \exists x Rxa,\ \forall x \forall y Ryx\}$
3. $\{\neg \exists x \exists y Lxy,\ Laa\}$
4. $\{\forall x(Px \rightarrow Qx),\ \forall z(Pz \rightarrow Rz),\ \forall y Py,\ \neg Qa \,\&\, \neg Rb\}$

$\star$ **Part Q** Write a symbolization key for the following argument, translate it, and prove it:

> There is someone who likes everyone who likes everyone that he likes. Therefore, there is someone who likes himself.

Part R Provide a proof of each claim.

1. $\{Pa \vee Qb, Qb \rightarrow b = c, \neg Pa\} \vdash Qc$
2. $\{m = n \vee n = o, An\} \vdash Am \vee Ao$
3. $\{\forall x x = m, Rma\} \vdash \exists x Rxx$
4. $\neg \exists x x \neq m \vdash \forall x \forall y (Px \rightarrow Py)$
5. $\forall x \forall y (Rxy \rightarrow x = y) \vdash Rab \rightarrow Rba$
6. $\{\exists x Jx, \exists x \neg Jx\} \vdash \exists x \exists y\ x \neq y$
7. $\{\forall x (x = n \leftrightarrow Mx), \forall x (Ox \vee \neg Mx)\} \vdash On$
8. $\{\exists x Dx, \forall x (x = p \leftrightarrow Dx)\} \vdash Dp$
9. $\{\exists x \big[Kx \,\&\, \forall y (Ky \rightarrow x = y) \,\&\, Bx\big], Kd\} \vdash Bd$
10. $\vdash Pa \rightarrow \forall x (Px \vee x \neq a)$

Part S Look back at Part D on p. 76. For each argument: If it is valid in QL, give a proof. If it is invalid, construct a model to show that it is invalid.

⋆ **Part T** For each of the following pairs of sentences: If they are logically equivalent in QL, give proofs to show this. If they are not, construct a model to show this.

1. $\forall x Px \rightarrow Qc$, $\forall x (Px \rightarrow Qc)$
2. $\forall x Px \,\&\, Qc$, $\forall x (Px \,\&\, Qc)$
3. $Qc \vee \exists x Qx$, $\exists x (Qc \vee Qx)$
4. $\forall x \forall y \forall z Bxyz$, $\forall x Bxxx$
5. $\forall x \forall y Dxy$, $\forall y \forall x Dxy$
6. $\exists x \forall y Dxy$, $\forall y \exists x Dxy$

⋆ **Part U** For each of the following arguments: If it is valid in QL, give a proof. If it is invalid, construct a model to show that it is invalid.

1. $\forall x \exists y Rxy$, $\therefore \exists y \forall x Rxy$
2. $\exists y \forall x Rxy$, $\therefore \forall x \exists y Rxy$
3. $\exists x (Px \,\&\, \neg Qx)$, $\therefore \forall x (Px \rightarrow \neg Qx)$
4. $\forall x (Sx \rightarrow Ta)$, Sd, $\therefore Ta$
5. $\forall x (Ax \rightarrow Bx)$, $\forall x (Bx \rightarrow Cx)$, $\therefore \forall x (Ax \rightarrow Cx)$
6. $\exists x (Dx \vee Ex)$, $\forall x (Dx \rightarrow Fx)$, $\therefore \exists x (Dx \,\&\, Fx)$
7. $\forall x \forall y (Rxy \vee Ryx)$, $\therefore Rjj$
8. $\exists x \exists y (Rxy \vee Ryx)$, $\therefore Rjj$
9. $\forall x Px \rightarrow \forall x Qx$, $\exists x \neg Px$, $\therefore \exists x \neg Qx$
10. $\exists x Mx \rightarrow \exists x Nx$, $\neg \exists x Nx$, $\therefore \forall x \neg Mx$

Part V

1. If you know that $\mathcal{A} \vdash \mathcal{B}$, what can you say about $(\mathcal{A} \,\&\, \mathcal{C}) \vdash \mathcal{B}$? Explain your answer.
2. If you know that $\mathcal{A} \vdash \mathcal{B}$, what can you say about $(\mathcal{A} \vee \mathcal{C}) \vdash \mathcal{B}$? Explain your answer.

Chapter A

Symbolic notation

In the history of formal logic, different symbols have been used at different times and by different authors. Often, authors were forced to use notation that their printers could typeset.

In one sense, the symbols used for various logical constants is arbitrary. There is nothing written in heaven that says that '$\neg$' must be the symbol for truth-functional negation. We might have specified a different symbol to play that part. Once we have given definitions for well-formed formulae (wff) and for truth in our logic languages, however, using '$\neg$' is no longer arbitrary. That is the symbol for negation in this textbook, and so it is the symbol for negation when writing sentences in our languages SL or QL.

summary of symbols	
negation	$\neg$, $\sim$
conjunction	$\&$, $\wedge$, $\bullet$
disjunction	$\vee$
conditional	$\rightarrow$, $\supset$
biconditional	$\leftrightarrow$, $\equiv$

This appendix presents some common symbols, so that you can recognize them if you encounter them in an article or in another book.

Negation Two commonly used symbols are the *hoe*, '$\neg$', and the *swung dash*, '$\sim$.' In some more advanced formal systems it is necessary to distinguish between two kinds of negation; the distinction is sometimes represented by using both '$\neg$' and '$\sim$.'

Disjunction The symbol '$\vee$' is typically used to symbolize inclusive disjunction.

Conjunction Conjunction is often symbolized with the *ampersand*, '&.' The ampersand is actually a decorative form of the Latin word 'et' which means 'and'; it is commonly used in English writing. As a symbol in a formal system, the ampersand is not the word 'and'; its meaning is given by the formal semantics for the language. Perhaps to avoid this confusion, some systems use a different symbol for conjunction. For example, '$\wedge$' is a counterpart to the symbol used for disjunction. Sometimes a single dot, '$\bullet$', is used. In some older texts, there is no symbol for conjunction at all; 'A and B' is simply written 'AB.'

Material Conditional There are two common symbols for the material conditional: the *arrow*, '$\rightarrow$', and the *hook*, '$\supset$.'

Material Biconditional The *double-headed arrow*, '$\leftrightarrow$', is used in systems that use the arrow to represent the material conditional. Systems that use the hook for the conditional typically use the *triple bar*, '$\equiv$', for the biconditional.

Quantifiers The universal quantifier is typically symbolized as an upside-down A, '$\forall$', and the existential quantifier as a backwards E, '$\exists$.' In some texts, there is no separate symbol for the universal quantifier. Instead, the variable is just written in parentheses in front of the formula that it binds. For example, 'all x are P' is written $(x)Px$.

In some systems, the quantifiers are symbolized with larger versions of the symbols used for conjunction and disjunction. Although quantified expressions cannot be translated into expressions without quantifiers, there is a conceptual connection between the universal quantifier and conjunction and between the existential quantifier and disjunction. Consider the sentence $\exists x Px$, for example. It means that *either* the first member of the UD is a P, *or* the second one is, *or* the third one is, ... Such a system uses the symbol '$\bigvee$' instead of '$\exists$.'

Polish notation

This section briefly discusses sentential logic in Polish notation, a system of notation introduced in the late 1920s by the Polish logician Jan Łukasiewicz.

Lower case letters are used as sentence letters. The capital letter N is used for negation. A is used for disjunction, K for conjunction, C for the conditional, E for the biconditional. ('A' is for alternation, another name for logical disjunction. 'E' is for equivalence.)

notation of SL	Polish notation
$\neg$	N
$\&$	K
$\vee$	A
$\rightarrow$	C
$\leftrightarrow$	E

In Polish notation, a binary connective is written *before* the two sentences that it connects. For example, the sentence $A \& B$ of SL would be written Kab in Polish notation.

The sentences $\neg A \rightarrow B$ and $\neg(A \rightarrow B)$ are very different; the main logical operator of the first is the conditional, but the main connective of the second is negation. In SL, we show this by putting parentheses around the conditional in the second sentence. In Polish notation, parentheses are never required. The left-most connective is always the main connective. The first sentence would simply be written $CNab$ and the second $NCab$.

This feature of Polish notation means that it is possible to evaluate sentences simply by working through the symbols from right to left. If you were constructing a truth table for $NKab$, for example, you would first consider the truth-values assigned to b and a, then consider their conjunction, and then negate the result. The general rule for what to evaluate next in SL is not nearly so simple. In SL, the truth table for $\neg(A \,\&\, B)$ requires looking at A and B, then looking in the middle of the sentence at the conjunction, and then at the beginning of the sentence at the negation. Because the order of operations can be specified more mechanically in Polish notation, variants of Polish notation are used as the internal structure for many computer programming languages.

Chapter B

Solutions to selected exercises

Many of the exercises may be answered correctly in different ways. Where that is the case, the solution here represents one possible correct answer.

Chapter 1 Part C

1. consistent
2. inconsistent
3. consistent
4. consistent

Chapter 1 Part D 1, 2, 3, 6, 8, and 10 are possible.

Chapter 2 Part A

1. $\neg M$
2. $M \vee \neg M$
3. $G \vee C$
4. $\neg C \,\&\, \neg G$
5. $C \rightarrow (\neg G \,\&\, \neg M)$
6. $M \vee (C \vee G)$

Chapter 2 Part C

1. $E_1 \,\&\, E_2$
2. $F_1 \rightarrow S_1$
3. $F_1 \vee E_1$
4. $E_2 \,\&\, \neg S_2$

5. $\neg E_1 \,\&\, \neg E_2$
6. $E_1 \,\&\, E_2 \,\&\, \neg(S_1 \vee S_2)$
7. $S_2 \rightarrow F_2$
8. $(\neg E_1 \rightarrow \neg E_2) \,\&\, (E_1 \rightarrow E_2)$
9. $S_1 \leftrightarrow \neg S_2$
10. $(E_2 \,\&\, F_2) \rightarrow S_2$
11. $\neg(E_2 \,\&\, F_2)$
12. $(F_1 \,\&\, F_2) \leftrightarrow (\neg E_1 \,\&\, \neg E_2)$

Chapter 2 Part D

A: Alice is a spy.
B: Bob is a spy.
C: The code has been broken.
G: The German embassy will be in an uproar.

1. $A \,\&\, B$
2. $(A \vee B) \rightarrow C$
3. $\neg(A \vee B) \rightarrow \neg C$
4. $G \vee C$
5. $(C \vee \neg C) \,\&\, G$
6. $(A \vee B) \,\&\, \neg(A \,\&\, B)$

Chapter 2 Part G

1. (a) no (b) no
2. (a) no (b) yes
3. (a) yes (b) yes
4. (a) no (b) no
5. (a) yes (b) yes
6. (a) no (b) no
7. (a) no (b) yes
8. (a) no (b) yes
9. (a) no (b) no

Chapter 3 Part A

1. tautology
2. contradiction
3. contingent
4. tautology
5. tautology
6. contingent
7. tautology
8. contradiction

9. tautology
10. contradiction
11. tautology
12. contingent
13. contradiction
14. contingent
15. tautology
16. tautology
17. contingent
18. contingent

Chapter 3 Part B 2, 3, 5, 6, 8, and 9 are logically equivalent.

Chapter 3 Part C 1, 3, 6, 7, and 8 are consistent.

Chapter 3 Part D 3, 5, 8, and 10 are valid.

Chapter 3 Part E

1. $\mathcal{A}$ and $\mathcal{B}$ have the same truth value on every line of a complete truth table, so $\mathcal{A} \leftrightarrow \mathcal{B}$ is true on every line. It is a tautology.
2. The sentence is false on some line of a complete truth table. On that line, $\mathcal{A}$ and $\mathcal{B}$ are true and $\mathcal{C}$ is false. So the argument is invalid.
3. Since there is no line of a complete truth table on which all three sentences are true, the conjunction is false on every line. So it is a contradiction.
4. Since $\mathcal{A}$ is false on every line of a complete truth table, there is no line on which $\mathcal{A}$ and $\mathcal{B}$ are true and $\mathcal{C}$ is false. So the argument is valid.
5. Since $\mathcal{C}$ is true on every line of a complete truth table, there is no line on which $\mathcal{A}$ and $\mathcal{B}$ are true and $\mathcal{C}$ is false. So the argument is valid.
6. Not much. $(\mathcal{A} \vee \mathcal{B})$ is a tautology if $\mathcal{A}$ and $\mathcal{B}$ are tautologies; it is a contradiction if they are contradictions; it is contingent if they are contingent.
7. $\mathcal{A}$ and $\mathcal{B}$ have different truth values on at least one line of a complete truth table, and $(\mathcal{A} \vee \mathcal{B})$ will be true on that line. On other lines, it might be true or false. So $(\mathcal{A} \vee \mathcal{B})$ is either a tautology or it is contingent; it is *not* a contradiction.

Chapter 3 Part F

1. $\neg A \rightarrow B$
2. $\neg(A \rightarrow \neg B)$
3. $\neg[(A \rightarrow B) \rightarrow \neg(B \rightarrow A)]$

Chapter 4 Part A

1. $Za \,\&\, Zb \,\&\, Zc$

2. $Rb \,\&\, \neg Ab$
3. $Lcb \to Mb$
4. $(Ab \,\&\, Ac) \to (Lab \,\&\, Lac)$
5. $\exists x(Rx \,\&\, Zx)$
6. $\forall x(Ax \to Rx)$
7. $\forall x\big[Zx \to (Mx \vee Ax)\big]$
8. $\exists x(Rx \,\&\, \neg Ax)$
9. $\exists x(Rx \,\&\, Lcx)$
10. $\forall x\big[(Mx \,\&\, Zx) \to Lbx\big]$
11. $\forall x\big[(Mx \,\&\, Lax) \to Lxa\big]$
12. $\exists xRx \to Ra$
13. $\forall x(Ax \to Rx)$
14. $\forall x\big[(Mx \,\&\, Lcx) \to Lax\big]$
15. $\exists x(Mx \,\&\, Lxb \,\&\, \neg Lbx)$

Chapter 4 Part E

1. $\neg\exists xTx$
2. $\forall x(Mx \to Sx)$
3. $\exists x\neg Sx$
4. $\exists x[Cx \,\&\, \neg\exists yByx]$
5. $\neg\exists xBxx$
6. $\neg\exists x(Cx \,\&\, \neg Sx \,\&\, Tx)$
7. $\exists x(Cx \,\&\, Tx) \,\&\, \exists x(Mx \,\&\, Tx) \,\&\, \neg\exists x(Cx \,\&\, Mx \,\&\, Tx)$
8. $\forall x[Cx \to \forall y(\neg Cy \to Bxy)]$
9. $\forall x\big((Cx \,\&\, Mx) \to \forall y[(\neg Cy \,\&\, \neg My) \to Bxy]\big)$

Chapter 4 Part G

1. $\forall x(Cxp \to Dx)$
2. $Cjp \,\&\, Fj$
3. $\exists x(Cxp \,\&\, Fx)$
4. $\neg\exists xSxj$
5. $\forall x\big[(Cxp \,\&\, Fx) \to Dx\big]$
6. $\neg\exists x(Cxp \,\&\, Mx)$
7. $\exists x(Cjx \,\&\, Sxe \,\&\, Fj)$
8. $Spe \,\&\, Mp$
9. $\forall x\big[(Sxp \,\&\, Mx) \to \neg\exists yCyx\big]$
10. $\exists x(Sxj \,\&\, \exists yCyx \,\&\, Fj)$
11. $\forall x\big[Dx \to \exists y(Sxy \,\&\, Fy \,\&\, Dy)\big]$
12. $\forall x\big[(Mx \,\&\, Dx) \to \exists y(Cxy \,\&\, Dy)\big]$

Chapter 4 Part J

1. $\forall x(Cx \to Bx)$

2. $\neg\exists x Wx$
3. $\exists x\exists y(Cx \,\&\, Cy \,\&\, x \neq y)$
4. $\exists x\exists y(Jx \,\&\, Ox \,\&\, Jy \,\&\, Oy \,\&\, x \neq y)$
5. $\forall x\forall y\forall z\big[(Jx \,\&\, Ox \,\&\, Jy \,\&\, Oy \,\&\, Jz \,\&\, Oz) \rightarrow (x = y \vee x = z \vee y = z)\big]$
6. $\exists x\exists y\big(Jx \,\&\, Bx \,\&\, Jy \,\&\, By \,\&\, x \neq y \,\&\, \forall z[(Jz \,\&\, Bz) \rightarrow (x = z \vee y = z)]\big)$
7. $\exists x_1\exists x_2\exists x_3\exists x_4\big[Dx_1 \,\&\, Dx_2 \,\&\, Dx_3 \,\&\, Dx_4 \,\&\, x_1 \neq x_2 \,\&\, x_1 \neq x_3 \,\&\, x_1 \neq x_4 \,\&\, x_2 \neq x_3 \,\&\, x_2 \neq x_4 \,\&\, x_3 \neq x_4 \,\&\, \neg\exists y(Dy \,\&\, y \neq x_1 \,\&\, y \neq x_2 \,\&\, y \neq x_3 \,\&\, y \neq x_4)\big]$
8. $\exists x\big(Dx \,\&\, Cx \,\&\, \forall y[(Dy \,\&\, Cy) \rightarrow x = y] \,\&\, Bx\big)$
9. $\forall x\big[(Ox \,\&\, Jx) \rightarrow Wx\big] \,\&\, \exists x\big[Mx \,\&\, \forall y(My \rightarrow x = y) \,\&\, Wx\big]$
10. $\exists x\big(Dx \,\&\, Cx \,\&\, \forall y[(Dy \,\&\, Cy) \rightarrow x = y] \,\&\, Wx\big) \rightarrow \exists x\forall y(Wx \leftrightarrow x = y)$
11. wide scope: $\neg\exists x\big[Mx \,\&\, \forall y(My \rightarrow x = y) \,\&\, Jx\big]$
    narrow scope: $\exists x\big[Mx \,\&\, \forall y(My \rightarrow x = y) \,\&\, \neg Jx\big]$
12. wide scope: $\neg\exists x\exists z\big(Dx \,\&\, Cx \,\&\, Mz \,\&\, \forall y[(Dy \,\&\, Cy) \rightarrow x = y] \,\&\, \forall y[(My \rightarrow z = y) \,\&\, x = z]\big)$
    narrow scope: $\exists x\exists z\big(Dx \,\&\, Cx \,\&\, Mz \,\&\, \forall y[(Dy \,\&\, Cy) \rightarrow x = y] \,\&\, \forall y[(My \rightarrow z = y) \,\&\, x \neq z]\big)$

Chapter 5 Part A 2, 3, 4, 6, 8, and 9 are true in the model.

Chapter 5 Part B 4, 5, and 7 are true in the model.

Chapter 5 Part D

$$\begin{aligned}
\text{UD} &= \{10,11,12,13\}\\
\text{extension}(O) &= \{11,13\}\\
\text{extension}(S) &= \emptyset\\
\text{extension}(T) &= \{10,11,12,13\}\\
\text{extension}(U) &= \{13\}\\
\text{extension}(N) &= \{<11,10>,<12,11>,<13,12>\}
\end{aligned}$$

Chapter 5 Part E

1. The sentence is true in this model:
$$\begin{aligned}
\text{UD} &= \{\text{Stan}\}\\
\text{extension}(D) &= \{\text{Stan}\}\\
\text{referent}(a) &= \text{Stan}\\
\text{referent}(b) &= \text{Stan}
\end{aligned}$$
   And it is false in this model:
$$\begin{aligned}
\text{UD} &= \{\text{Stan}\}\\
\text{extension}(D) &= \emptyset\\
\text{referent}(a) &= \text{Stan}\\
\text{referent}(b) &= \text{Stan}
\end{aligned}$$
2. The sentence is true in this model:
$$\begin{aligned}
\text{UD} &= \{\text{Stan}\}\\
\text{extension}(T) &= \{<\text{Stan, Stan}>\}\\
\text{referent}(h) &= \text{Stan}
\end{aligned}$$

And it is false in this model:

$$\begin{aligned}\text{UD} &= \{\text{Stan}\}\\ \text{extension}(T) &= \emptyset\\ \text{referent}(h) &= \text{Stan}\end{aligned}$$

3. The sentence is true in this model:

$$\begin{aligned}\text{UD} &= \{\text{Stan, Ollie}\}\\ \text{extension}(P) &= \{\text{Stan}\}\\ \text{referent}(m) &= \text{Stan}\end{aligned}$$

And it is false in this model:

$$\begin{aligned}\text{UD} &= \{\text{Stan}\}\\ \text{extension}(P) &= \emptyset\\ \text{referent}(m) &= \text{Stan}\end{aligned}$$

Chapter 5 Part F There are many possible correct answers. Here are some:

1. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{alpha}\}\\ \text{extension}(J) &= \{\text{alpha}\}\\ \text{extension}(K) &= \emptyset\\ \text{referent}(a) &= \text{alpha}\end{aligned}$$

2. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(J) &= \{\text{alpha}\}\\ \text{referent}(m) &= \text{omega}\end{aligned}$$

3. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(R) &= \{<\text{alpha,alpha}>\}\end{aligned}$$

4. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(P) &= \{\text{alpha}\}\\ \text{extension}(Q) &= \emptyset\\ \text{referent}(c) &= \text{alpha}\end{aligned}$$

5. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{iota}\}\\ \text{extension}(P) &= \emptyset\\ \text{extension}(Q) &= \emptyset\end{aligned}$$

6. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{iota}\}\\ \text{extension}(P) &= \emptyset\\ \text{extension}(Q) &= \{\text{iota}\}\end{aligned}$$

7. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{iota}\}\\ \text{extension}(P) &= \emptyset\\ \text{extension}(Q) &= \{\text{iota}\}\end{aligned}$$

8. Making the first sentence true and the second false:

   UD = {alpha, omega}
   extension(R) = {<alpha, omega>, <omega, alpha>}
9. Making the first sentence false and the second true:

   UD = {alpha, omega}
   extension(R) = {<alpha, alpha>, <alpha, omega>}

Chapter 5 Part I

1. There are many possible answers. Here is one:

   UD = {Harry, Sally}
   extension(R) = {<Sally, Harry>}
   referent(a) = Harry
2. There are no predicates or constants, so we only need to give a UD. Any UD with 2 members will do.
3. We need to show that it is impossible to construct a model in which these are both true. Suppose $\exists x\, x \neq a$ is true in a model. There is something in the universe of discourse that is *not* the referent of a. So there are at least two things in the universe of discourse: referent(a) and this other thing. Call this other thing β— we know $a \neq \beta$. But if $a \neq \beta$, then $\forall x \forall y\, x = y$ is false. So the first sentence must be false if the second sentence is true. As such, there is no model in which they are both true. Therefore, they are inconsistent.

Chapter 5 Part J

2. No, it would not make any difference. The satisfaction of a formula with one or more free variables depends on what the variable assignment does for those variables. Because a sentence has no free variables, however, its satisfaction does not depend on the variable assignment. So a sentence that is satisfied by *some* variable assignment is satisfied by *every* other variable assignment as well.

Chapter 6 Part A

1	$W \to \neg B$	
2	$A \,\&\, W$	
3	$B \vee (J \,\&\, K)$	
4	W	&E 2
5	$\neg B$	→E 1, 4
6	$J \,\&\, K$	∨E 3, 5
7	K	&E 6

1	$L \leftrightarrow \neg O$	
2	$L \vee \neg O$	
3	$\quad \neg L$	
4	$\quad \neg O$	$\vee$E 2, 3
5	$\quad L$	$\leftrightarrow$E 1, 4
6	$\quad \neg L$	R 3
7	L	$\neg$E 3–6

1	$Z \rightarrow (C \,\&\, \neg N)$	
2	$\neg Z \rightarrow (N \,\&\, \neg C)$	
3	$\quad \neg(N \vee C)$	
4	$\quad \neg N \,\&\, \neg C$	DeM 3
5	$\quad\quad Z$	
6	$\quad\quad C \,\&\, \neg N$	$\rightarrow$E 1, 5
7	$\quad\quad C$	&E 6
8	$\quad\quad \neg C$	&E 4
9	$\quad \neg Z$	$\neg$I 5–8
10	$\quad N \,\&\, \neg C$	$\rightarrow$E 2, 9
11	$\quad N$	&E 10
12	$\quad \neg N$	&E 4
13	$N \vee C$	$\neg$E 3–12

Chapter 6 Part B

1.

1	$K \,\&\, L$	want $K \leftrightarrow L$
2	$\quad K$	want L
3	$\quad L$	&E 1
4	$\quad L$	want K
5	$\quad K$	&E 1
6	$K \leftrightarrow L$	$\leftrightarrow$I 2–3, 4–5

2.

1	$A \rightarrow (B \rightarrow C)$	want $(A \,\&\, B) \rightarrow C$
2	$\quad A \,\&\, B$	want C
3	$\quad A$	&E 2
4	$\quad B \rightarrow C$	$\rightarrow$E 1, 3
5	$\quad B$	&E 2
6	$\quad C$	$\rightarrow$E 4, 5
7	$(A \,\&\, B) \rightarrow C$	$\rightarrow$I 2–6

3.

1	$P \,\&\, (Q \vee R)$	
2	$P \to \neg R$	want $Q \vee E$
3	P	&E 1
4	$\neg R$	$\to$E 2, 3
5	$Q \vee R$	&E 1
6	Q	$\vee$E 5, 4
7	$Q \vee E$	$\vee$I 6

4.

1	$(C \,\&\, D) \vee E$	want $E \vee D$
2	$\quad \neg E$	want D
3	$\quad C \,\&\, D$	$\vee$E 1, 2
4	$\quad D$	&E 3
5	$\neg E \to D$	$\to$I 2–4
6	$E \vee D$	MC 5

5.

1	$\neg F \to G$	
2	$F \to H$	want $G \vee H$
3	$\quad \neg G$	want H
4	$\quad \neg\neg F$	MT 1, 3
5	$\quad F$	DN 4
6	$\quad H$	$\to$E 2, 5
7	$\neg G \to H$	$\to$I 3–6
8	$G \vee H$	MC 7

6.

1	$(X \,\&\, Y) \vee (X \,\&\, Z)$	
2	$\neg(X \,\&\, D)$	
3	$D \vee M$	want M
4	$\quad \neg X$	for reductio
5	$\quad \neg X \vee \neg Y$	$\vee$I 4
6	$\quad \neg(X \,\&\, Y)$	DeM 5
7	$\quad X \,\&\, Z$	$\vee$E 1, 6
8	$\quad X$	&E 7
9	$\quad \neg X$	R 4
10	X	$\neg$E 4–9
11	$\quad \neg M$	for reductio
12	$\quad D$	$\vee$E 3, 11
13	$\quad X \,\&\, D$	&I 10, 12
14	$\quad \neg(X \,\&\, D)$	R 2
15	M	$\neg$E 11–14

Chapter 6 Part H

1. Rca, Rcb, Rcc, and Rcd are substitution instances of $\forall x Rcx$.
2. Of the expressions listed, only $\forall y Lby$ is a substitution instance of $\exists x \forall y Lxy$.

Chapter 6 Part I

1	$\forall x \exists y (Rxy \vee Ryx)$	
2	$\forall x \neg Rmx$	
3	$\exists y (Rmy \vee Rym)$	$\forall$E 1
4	$\quad Rma \vee Ram$	
5	$\quad \neg Rma$	$\forall$E 2
6	$\quad Ram$	$\vee$E 4, 5
7	$\quad \exists x Rxm$	$\exists$I 6
8	$\exists x Rxm$	$\exists$E 3, 4–7

1	$\forall x (\exists y Lxy \rightarrow \forall z Lzx)$	
2	Lab	
3	$\exists y Lay \rightarrow \forall z Lza$	$\forall$E 1
4	$\exists y Lay$	$\exists$I 2
5	$\forall z Lza$	$\rightarrow$E 3, 4
6	Lca	$\forall$E 5
7	$\exists y Lcy \rightarrow \forall z Lzc$	$\forall$E 1
8	$\exists y Lcy$	$\exists$I 6
9	$\forall z Lzc$	$\rightarrow$E 7, 8
10	Lcc	$\forall$E 9
11	$\forall x Lxx$	$\forall$I 10

1	$\forall x(Jx \to Kx)$	
2	$\exists x \forall y Lxy$	
3	$\forall x Jx$	
4	$\forall y Lay$	
5	Ja	$\forall$E 3
6	$Ja \to Ka$	$\forall$E 1
7	Ka	$\to$E 6, 5
8	Laa	$\forall$E 4
9	$Ka \,\&\, Laa$	&I 7, 8
10	$\exists x(Kx \,\&\, Lxx)$	$\exists$I 9
11	$\exists x(Kx \,\&\, Lxx)$	$\exists$E 2, 4–10

1	$\neg(\exists x Mx \vee \forall x \neg Mx)$	
2	$\neg\exists x Mx \,\&\, \neg\forall x \neg Mx$	DeM 1
3	$\neg\exists x Mx$	&E 2
4	$\forall x \neg Mx$	QN 3
5	$\neg\forall x \neg Mx$	&E 2
6	$\exists x Mx \vee \forall x \neg Mx$	$\neg$E 1–5

Chapter 6 Part J

1.

1	$\neg(\forall x Fx \vee \neg\forall x Fx)$	for reductio
2	$\neg\forall x Fx \,\&\, \neg\neg\forall x Fx$	DeM 1
3	$\neg\forall x Fx$	&E 2
4	$\neg\neg\forall x Fx$	&E 2
5	$\forall x Fx \vee \neg\forall x Fx$	$\neg$E 1–4

2.

1	$\forall x(Mx \leftrightarrow Nx)$	
2	$Ma \,\&\, \exists x Rxa$	want $\exists x Nx$
3	$Ma \leftrightarrow Na$	$\forall$E 1
4	Ma	&E 2
5	Na	$\leftrightarrow$E 3, 4
6	$\exists x Nx$	$\exists$I 5

3.

1	$\forall x(\neg Mx \vee Ljx)$	
2	$\forall x(Bx \rightarrow Ljx)$	
3	$\forall x(Mx \vee Bx)$	want $\forall x Ljx$
4	$\neg Ma \vee Lja$	$\forall$E 1
5	$Ma \rightarrow Lja$	MC 4
6	$Ba \rightarrow Lja$	$\forall$E 2
7	$Ma \vee Ba$	$\forall$E 3
8	Lja	DIL 7, 5, 6
9	$\forall x Ljx$	$\forall$I 8

4.

1	$\forall x(Cx \,\&\, Dt)$	want $\forall x Cx \,\&\, Dt$
2	$Ca \,\&\, Dt$	$\forall$E 1
3	Ca	&E 2
4	$\forall x Cx$	$\forall$I 3
5	Dt	&E 2
6	$\forall x Cx \,\&\, Dt$	&I 4, 5

5.

1	$\exists x(Cx \vee Dt)$	want $\exists x Cx \vee Dt$
2	$\quad Ca \vee Dt$	for $\exists$E
3	$\quad\quad \neg(\exists x Cx \vee Dt)$	for reductio
4	$\quad\quad \neg\exists x Cx \,\&\, \neg Dt$	DeM 3
5	$\quad\quad \neg Dt$	&E 4
6	$\quad\quad Ca$	$\vee$E 2, 5
7	$\quad\quad \exists x Cx$	$\exists$I 6
8	$\quad\quad \neg\exists x Cx$	&E 4
9	$\quad \exists x Cx \vee Dt$	$\neg$E 3–8
10	$\exists x Cx \vee Dt$	$\exists$E 1, 2–9

Chapter 6 Part Q Regarding the translation of this argument, see p. 66.

1	$\exists x \forall y[\forall z(Lxz \to Lyz) \to Lxy]$	
2	$\quad \forall y[\forall z(Laz \to Lyz) \to Lay]$	
3	$\quad \forall z(Laz \to Laz) \to Laa$	∀E 2
4	$\quad\quad \neg\exists x Lxx$	for reductio
5	$\quad\quad \forall x \neg Lxx$	QN 4
6	$\quad\quad \neg Laa$	∀E 5
7	$\quad\quad \neg\forall z(Laz \to Laz)$	MT 5, 6
8	$\quad\quad\quad Lab$	
9	$\quad\quad\quad Lab$	R 8
10	$\quad\quad Lab \to Lab$	→I 8—9
11	$\quad\quad \forall z(Laz \to Laz)$	∀I 10
12	$\quad\quad \neg\forall z(Laz \to Laz)$	R 7
13	$\quad \exists x Lxx$	¬E 4—12
14	$\exists x Lxx$	∃E 1, 2—13

Chapter 6 Part T 2, 3, and 5 are logically equivalent.

Chapter 6 Part U 2, 4, 5, 7, and 10 are valid. Here are complete answers for some of them:

1. UD = {mocha, freddo}
   extension(R) = {<mocha, freddo>, <freddo, mocha>}

2.

1	$\exists y \forall x Rxy$	want $\forall x \exists y Rxy$
2	$\quad \forall x Rxa$	
3	$\quad Rba$	∀E 2
4	$\quad \exists y Rby$	∃I 3
5	$\quad \forall x \exists y Rxy$	∀I 4
6	$\forall x \exists y Rxy$	∃E 1, 2–5

Quick Reference

$\mathcal{A}$	$\neg\mathcal{A}$
T	F
F	T

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\&\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$	$\mathcal{A}\leftrightarrow\mathcal{B}$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

$\mathcal{A}$	$\neg\mathcal{A}$
1	0
0	1

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\&\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$	$\mathcal{A}\leftrightarrow\mathcal{B}$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

Symbolization

Sentential Connectives (chapter 2)

It is not the case that P.	$\neg P$
Either P, or Q.	$(P \vee Q)$
Neither P, nor Q.	$\neg(P \vee Q)$ or $(\neg P \& \neg Q)$
Both P, and Q.	$(P \& Q)$
If P, then Q.	$(P \rightarrow Q)$
P only if Q.	$(P \rightarrow Q)$
P if and only if Q.	$(P \leftrightarrow Q)$
Unless P, Q. P unless Q.	$(P \vee Q)$

Predicates (chapter 4)

All Fs are Gs.	$\forall x(Fx \rightarrow Gx)$
Some Fs are Gs.	$\exists x(Fx \& Gx)$
Not all Fs are Gs.	$\neg\forall x(Fx \rightarrow Gx)$ or $\exists x(Fx \& \neg Gx)$
No Fs are Gs.	$\forall x(Fx \rightarrow \neg Gx)$ or $\neg\exists x(Fx \& Gx)$

Identity (section 4.6)

Only j is G.	$\forall x(Gx \leftrightarrow x = j)$
Everything besides j is G.	$\forall x(x \neq j \rightarrow Gx)$
The F is G.	$\exists x(Fx \& \forall y(Fy \rightarrow x = y) \& Gx)$

'The F is not G' can be translated two ways:

It is not the case that the F is G. (wide)	$\neg\exists x(Fx \& \forall y(Fy \rightarrow x = y) \& Gx)$
The F is non-G. (narrow)	$\exists x(Fx \& \forall y(Fy \rightarrow x = y) \& \neg Gx)$

Using identity to symbolize quantities

There are at least _____ Fs.

one $\exists x Fx$
two $\exists x_1 \exists x_2 (Fx_1 \,\&\, Fx_2 \,\&\, x_1 \neq x_2)$
three $\exists x_1 \exists x_2 \exists x_3 (Fx_1 \,\&\, Fx_2 \,\&\, Fx_3 \,\&\, x_1 \neq x_2 \,\&\, x_1 \neq x_3 \,\&\, x_2 \neq x_3)$
four $\exists x_1 \exists x_2 \exists x_3 \exists x_4 (Fx_1 \,\&\, Fx_2 \,\&\, Fx_3 \,\&\, Fx_4 \,\&\, x_1 \neq x_2 \,\&\, x_1 \neq x_3 \,\&\, x_1 \neq x_4 \,\&\, x_2 \neq x_3 \,\&\, x_2 \neq x_4 \,\&\, x_3 \neq x_4)$
n $\exists x_1 \cdots \exists x_n (Fx_1 \,\&\, \cdots \,\&\, Fx_n \,\&\, x_1 \neq x_2 \,\&\, \cdots \,\&\, x_{n-1} \neq x_n)$

There are at most _____ Fs.

One way to say 'at most n things are F' is to put a negation sign in front of one of the symbolizations above and say $\neg$'at least $n+1$ things are F.' Equivalently:

one $\forall x_1 \forall x_2 \big[(Fx_1 \,\&\, Fx_2) \rightarrow x_1 = x_2\big]$
two $\forall x_1 \forall x_2 \forall x_3 \big[(Fx_1 \,\&\, Fx_2 \,\&\, Fx_3) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3)\big]$
three $\forall x_1 \forall x_2 \forall x_3 \forall x_4 \big[(Fx_1 \,\&\, Fx_2 \,\&\, Fx_3 \,\&\, Fx_4) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_1 = x_4 \vee x_2 = x_3 \vee x_2 = x_4 \vee x_3 = x_4)\big]$
n $\forall x_1 \cdots \forall x_{n+1} \big[(Fx_1 \,\&\, \cdots \,\&\, Fx_{n+1}) \rightarrow (x_1 = x_2 \vee \cdots \vee x_n = x_{n+1})\big]$

There are exactly _____ Fs.

One way to say 'exactly n things are F' is to conjoin two of the symbolizations above and say 'at least n things are F' & 'at most n things are F.' The following equivalent formulae are shorter:

zero $\forall x \neg Fx$
one $\exists x \big[Fx \,\&\, \neg\exists y (Fy \,\&\, x \neq y)\big]$
two $\exists x_1 \exists x_2 \big[Fx_1 \,\&\, Fx_2 \,\&\, x_1 \neq x_2 \,\&\, \neg\exists y \big(Fy \,\&\, y \neq x_1 \,\&\, y \neq x_2\big)\big]$
three $\exists x_1 \exists x_2 \exists x_3 \big[Fx_1 \,\&\, Fx_2 \,\&\, Fx_3 \,\&\, x_1 \neq x_2 \,\&\, x_1 \neq x_3 \,\&\, x_2 \neq x_3 \,\&\, \neg\exists y (Fy \,\&\, y \neq x_1 \,\&\, y \neq x_2 \,\&\, y \neq x_3)\big]$
n $\exists x_1 \cdots \exists x_n \big[Fx_1 \,\&\, \cdots \,\&\, Fx_n \,\&\, x_1 \neq x_2 \,\&\, \cdots \,\&\, x_{n-1} \neq x_n \,\&\, \neg\exists y (Fy \,\&\, y \neq x_1 \,\&\, \cdots \,\&\, y \neq x_n)\big]$

Specifying the size of the UD

Removing F from the symbolizations above produces sentences that talk about the size of the UD. For instance, 'there are at least 2 things (in the UD)' may be symbolized as $\exists x \exists y (x \neq y)$.

Basic Rules of Proof

Reiteration

m	$\mathcal{A}$	
	$\mathcal{A}$	R m

Conjunction Introduction

m	$\mathcal{A}$	
n	$\mathcal{B}$	
	$\mathcal{A} \,\&\, \mathcal{B}$	&I m, n

Conjunction Elimination

m	$\mathcal{A} \,\&\, \mathcal{B}$	
	$\mathcal{A}$	&E m

m	$\mathcal{A} \,\&\, \mathcal{B}$	
	$\mathcal{B}$	&E m

Disjunction Introduction

m	$\mathcal{A}$	
	$\mathcal{A} \vee \mathcal{B}$	$\vee$I m

m	$\mathcal{A}$	
	$\mathcal{B} \vee \mathcal{A}$	$\vee$I m

Disjunction Elimination

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\mathcal{A}$	$\vee$E m, n

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{A}$	
	$\mathcal{B}$	$\vee$E m, n

Conditional Introduction

m	$\quad\mathcal{A}$	want $\mathcal{B}$
n	$\quad\mathcal{B}$	
	$\mathcal{A} \to \mathcal{B}$	$\to$I m–n

Conditional Elimination

m	$\mathcal{A} \to \mathcal{B}$	
n	$\mathcal{A}$	
	$\mathcal{B}$	$\to$E m, n

Biconditional Introduction

m	$\quad\mathcal{A}$	want $\mathcal{B}$
n	$\quad\mathcal{B}$	
p	$\quad\mathcal{B}$	want $\mathcal{A}$
q	$\quad\mathcal{A}$	
	$\mathcal{A} \leftrightarrow \mathcal{B}$	$\leftrightarrow$I m–n, p–q

Biconditional Elimination

m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{B}$	
	$\mathcal{A}$	$\leftrightarrow$E m, n

m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{A}$	
	$\mathcal{B}$	$\leftrightarrow$E m, n

Negation Introduction

m	$\quad\mathcal{A}$	for reductio
$n-1$	$\quad\mathcal{B}$	
n	$\quad\neg\mathcal{B}$	
	$\neg\mathcal{A}$	$\neg$I m–n

Negation Elimination

m	$\quad\neg\mathcal{A}$	for reductio
$n-1$	$\quad\mathcal{B}$	
n	$\quad\neg\mathcal{B}$	
	$\mathcal{A}$	$\neg$E m–n

Quantifier Rules

Existential Introduction

$$
\begin{array}{r|ll}
m & \mathscr{A}c & \\
 & \exists \chi \mathscr{A}\chi & \exists\text{I } m
\end{array}
$$

Note that χ may replace some or all occurrences of c in $\mathscr{A}c$.

Existential Elimination

$$
\begin{array}{r|l l}
m & \exists \chi \mathscr{A}\chi & \\
n & \quad \begin{array}{|l} \mathscr{A}c^{*} \\ \hline \end{array} & \\
p & \quad \begin{array}{|l} \mathscr{B} \end{array} & \\
 & \mathscr{B} & \exists\text{E } m, n\text{–}p
\end{array}
$$

* c must not appear in $\exists \chi \mathscr{A}\chi$, in $\mathscr{B}$, or in any undischarged assumption.

Universal Introduction

$$
\begin{array}{r|ll}
m & \mathscr{A}c^{*} & \\
 & \forall \chi \mathscr{A}\chi & \forall\text{I } m
\end{array}
$$

* c must not occur in any undischarged assumptions.

Universal Elimination

$$
\begin{array}{r|ll}
m & \forall \chi \mathscr{A}\chi & \\
 & \mathscr{A}c & \forall\text{E } m
\end{array}
$$

Identity Rules

$$
\begin{array}{r|ll}
 & c = c & =\text{I}
\end{array}
$$

$$
\begin{array}{r|ll}
m & c = d & \\
n & \mathscr{A} & \\
 & \mathscr{A}c \circlearrowright d & =\text{E } m, n
\end{array}
$$

One constant may replace some or all occurrences of the other.

Derived Rules

Dilemma

$$
\begin{array}{r|ll}
m & \mathscr{A} \vee \mathscr{B} & \\
n & \mathscr{A} \rightarrow \mathscr{C} & \\
p & \mathscr{B} \rightarrow \mathscr{C} & \\
 & \mathscr{C} & \text{DIL } m, n, p
\end{array}
$$

Modus Tollens

$$
\begin{array}{r|ll}
m & \mathscr{A} \rightarrow \mathscr{B} & \\
n & \neg \mathscr{B} & \\
 & \neg \mathscr{A} & \text{MT } m, n
\end{array}
$$

Hypothetical Syllogism

$$
\begin{array}{r|ll}
m & \mathscr{A} \rightarrow \mathscr{B} & \\
n & \mathscr{B} \rightarrow \mathscr{C} & \\
 & \mathscr{A} \rightarrow \mathscr{C} & \text{HS } m, n
\end{array}
$$

Replacement Rules

Commutativity (Comm)

$$(\mathscr{A} \,\&\, \mathscr{B}) \iff (\mathscr{B} \,\&\, \mathscr{A})$$
$$(\mathscr{A} \vee \mathscr{B}) \iff (\mathscr{B} \vee \mathscr{A})$$
$$(\mathscr{A} \leftrightarrow \mathscr{B}) \iff (\mathscr{B} \leftrightarrow \mathscr{A})$$

DeMorgan (DeM)

$$\neg(\mathscr{A} \vee \mathscr{B}) \iff (\neg\mathscr{A} \,\&\, \neg\mathscr{B})$$
$$\neg(\mathscr{A} \,\&\, \mathscr{B}) \iff (\neg\mathscr{A} \vee \neg\mathscr{B})$$

Double Negation (DN)

$$\neg\neg\mathscr{A} \iff \mathscr{A}$$

Material Conditional (MC)

$$(\mathscr{A} \rightarrow \mathscr{B}) \iff (\neg\mathscr{A} \vee \mathscr{B})$$
$$(\mathscr{A} \vee \mathscr{B}) \iff (\neg\mathscr{A} \rightarrow \mathscr{B})$$

Biconditional Exchange ($\leftrightarrow$ex)

$$[(\mathscr{A} \rightarrow \mathscr{B}) \,\&\, (\mathscr{B} \rightarrow \mathscr{A})] \iff (\mathscr{A} \leftrightarrow \mathscr{B})$$

Quantifier Negation (QN)

$$\neg\forall\chi\mathscr{A} \iff \exists\chi\neg\mathscr{A}$$
$$\neg\exists\chi\mathscr{A} \iff \forall\chi\neg\mathscr{A}$$

In the Introduction to his volume *Symbolic Logic*, Charles Lutwidge Dodson advised: "When you come to any passage you don't understand, *read it again*: if you *still* don't understand it, *read it again*: if you fail, even after *three* readings, very likely your brain is getting a little tired. In that case, put the book away, and take to other occupations, and next day, when you come to it fresh, you will very likely find that it is *quite* easy."

The same might be said for this volume, although readers are forgiven if they take a break for snacks after *two* readings.

about the author:

P.D. Magnus is a professor of philosophy in Albany, New York. His primary research is in the philosophy of science.