Lógica Computacional

Introdução à lógica proposicional

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Segundo Machado e Cunha (2008) o objetivo fundamental de um curso de lógica é desenvolver a competência na argumentação, compreender as razões próprias e dos outros nas tomadas de posição diante dos acontecimentos e nas decisões. Mas afinal, o que esse desenvolvimento, que parece ser linguístico, tem a ver com nosso universo computacional? A verdade é que tem tudo a ver. Construiremos algoritmos capazes de tomar decisões, e para isso precisaremos implementar regras baseadas na Lógica Formal. Isso mesmo, aquela Lógica Formal desenvolvida por Aristóteles entre 300 e 400 anos antes de Cristo (MACHADO; CUNHA, 2008). Assim, nessa webaula vamos estudar algumas características para desenvolver as tomadas de decisões.

Premissas e conclusões

Para relembrarmos rapidamente, Aristóteles desenvolveu um método no qual ele separa a forma (podemos entender como regras) do conteúdo nas argumentações. Ou seja, no método Formal, "não são considerados os conteúdos das sentenças componentes de um argumento, mas apenas a forma de articulá-las ou o modo como umas são deduzidas das outras". (MACHADO; CUNHA, 2008, p. 15).

Na lógica computacional, vamos utilizar as mesmas regras da Lógica Formal, porém iremos valorar os conteúdos, como verdadeiro ou falso, a fim de extrair nossas conclusões.

Em nosso cotidiano, usamos a linguagem natural para nos expressar por meio de frases, que em alguns casos podem ser argumentativas sendo assim compostas por premissas e conclusões.

Vejamos um exemplo extraído de Machado e Cunha (2008, p. 16). Observe o argumento de uma professora sobre o desempenho de um certo aluno: "É lógico que Pedro será aprovado nos exames, pois ele é inteligente e estuda muito e todos os alunos inteligentes e estudiosos são aprovados". Esse argumento foi construído embasado por premissas (razões) e que levam a uma única conclusão.

Premissas (razões)	Conclusão
1. Pedro é inteligente.	Pedro será aprovado.
2. Pedro estuda muito.	
3. Todos os alunos inteligentes e estudiosos são aprovados.	

Veja que separamos a frase premissas e conclusão. Nesse caso, três premissas permitiram chegar a uma conclusão coerente. Extrair essa conclusão do argumento só foi possível devido às regras da lógica proposicional, que por meio de premissas e conectores extraem-se resultados lógicos. Fazer essa separação (premissa / conclusão) é muito importante, pois nem toda frase é um argumento.

Sentenças na lógica computacional: argumento e classificação

Para uma sentença ser argumento é preciso existir uma conclusão, logo, nem toda frase é um argumento. Por exemplo, a frase “Segure firme!”, não possui premissas e conclusões, pois trata-se de uma sentença imperativa (ordem) ou então a frase: “Você pode abrir a porta?” também não é um argumento, pois estamos diante de uma sentença interrogativa. As sentenças exclamativas, como por exemplo, “Que lindo!”, “Parabéns!” também não são consideradas argumentos. No estudo da lógica, além de distinguir se uma frase é ou não um argumento, também é importante distinguirmos se uma sentença pode ou não ser classificada como verdadeira ou falsa (não ambas ao mesmo tempo). Por exemplo, considere as frases:

1. O Brasil é um país da América Latina.
2. Minas Gerais é um estado do Nordeste.
3. São Paulo é a capital do Paraná.
4. Três mais um é igual a quatro.
5. Que horas são?

As quatro primeiras frases podem ser classificadas (valoradas) em verdadeira (V) ou falso (F). Veja:

1. Verdadeira.
2. Falso.
3. Falso.
4. Verdadeira.

Mas a quinta frase não pode ser valorada em V ou F, pois a resposta é um certo horário.

5. ?

Essa distinção entre os tipos de sentenças é crucial para o estudo da lógica, pois uma frase que pode ser classificada como verdadeira ou falsa (não ambas ao mesmo tempo) é chamada **proposição** (MACHADO; CUNHA, 2008; SOUZA, 2016; GERSTING, 2017).

> Proposição é uma sentença declarativa que pode ser classificada como verdadeira ou falsa, jamais ambas ao mesmo tempo. Ou seja, não pode haver dúvida quanto à classificação da sentença. Também podemos dizer que trata-se de uma classificação binária, pois só existem dois resultados possíveis: V ou F, ou ainda 1 ou 0.

Quando extraímos as proposições simples, podemos fazer adequações nos verbos, o mesmo acontece quando usamos proposições simples para fazer as compostas. Reescrevendo a frase utilizando a notação simbólica teremos: **Se** A **então** R. Nesse caso, temos duas palavras fazendo a ligação entre as proposições: Se... então.

Essas “palavras” usadas para unir as proposições simples são os conectivos (ou conectores) lógicos e influenciam a valoração de uma proposição composta. Os conectivos disponíveis para fazer a conexão são: (i) **e**, (ii) **ou**, (iii) **não**, (iv) **se... então**, (v) **se, e somente se**. (BISPO; CASTANHEIRA, 2011).

Conectivo lógico de conjunção - e

Vamos começar estudando o conectivo lógico “**e**”, que também pode ser visto na literatura escrito em inglês **AND** ou ainda por meio do seu símbolo $\wedge$. Essa operação lógica é chamada **conjunção** e sua valoração será verdadeira somente quando ambas as proposições simples forem verdadeiras. Resumindo, se A e B forem proposições simples verdadeiras, a proposição composta $A \wedge B$ (lê-se “A e B”) será verdadeira.

Conectivo lógico de disjunção – ou (inclusivo)

Outro conector muito utilizado tanto na Lógica Formal, quanto na computacional é “**ou**” que também pode ser visto na literatura escrito em inglês **OR** ou ainda por meio do seu símbolo $\vee$. Essa operação lógica é chamada de **disjunção** e seu operador lógico pode ser utilizado de duas formas distintas: **inclusivo** ou **exclusivo**.

O operador lógico de disjunção usado na forma inclusiva terá sua valoração falsa somente quando ambas as proposições simples forem falsas. Resumindo, se A e B forem proposições simples falsas, a proposição composta $A \vee B$ (lê-se "A ou B") será falsa, nos demais casos a valoração é verdadeira.

Operador lógico de negação – não

Os operadores lógicos de conjunção e disjunção são binários, ou seja, juntam duas expressões para formar uma nova proposição. O operador lógico de negação é unário, ou seja, ele não junta duas proposições, ele age sobre uma única proposição (que pode ser resultado de uma operação binária). A palavra usada para fazer a negação é o **não** que também pode ser visto na literatura em inglês **NOT**, ou ainda de forma simbólica como $\sim,\ \neg,\ '$. Os dois primeiros símbolos são usados antes da letra que representa a proposição, já o terceiro é usado depois da letra. Por exemplo, as expressões: $\sim A,\ \neg B,\ C'$ representam as negações das proposições $A,\ B,\ C$.

A operação lógica de negação troca o valor-verdade da proposição. Ou seja, se a proposição é verdadeira, quando acompanhada do operador de negação passará a ser falsa; por outro lado, se ela for falsa passará a ser verdadeira.

Nessa webaula estudamos algumas características da Lógica Formal. Continue estudando este conceito, pois sem essa lógica não conseguiríamos construir os algoritmos computacionais.

Lógica Computacional

Conectivos e classificação textual

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nessa webaula vamos estudar quatro conectivos lógicos que auxiliarão nos próximos estudos de lógica computacional.

Conectivos lógicos

Uma proposição composta pode ser criada fazendo a conjunção de duas proposições simples, neste caso são utilizadas as palavras “e”, “mas”, “no entanto”, dentre outras para fazer a conexão. Também podemos criar uma proposição composta fazendo a disjunção de duas proposições simples, nesse caso usamos a palavra “ou” para a conexão. A disjunção possui uma particularidade, ela pode ser inclusiva ou exclusiva.

Conectivo lógico de disjunção – ou (exclusivo)

Considere as seguintes proposições simples:

A. João é estudante.
B. João é trabalhador.
C. João é Paulista.
D. João é Carioca.

Agora vamos usar as proposições simples A, B, C, D, criar as compostas usando a disjunção. Observe:

R. João é estudante ou é trabalhador.
S. João é Paulista ou é Carioca.

A proposição R representa uma disjunção inclusiva, pois João pode ser estudante e também trabalhador. Já a proposição S é uma disjunção exclusiva, pois João não pode ser Paulista e Carioca, ele só pode ser um dos dois.

Segundo Souza (2016), a disjunção inclusiva é representada pelo símbolo $\vee$, ou seja, a proposição R, pode ser escrita como $A \vee B$. Já a disjunção exclusiva é representada pelo símbolo $\underline{\vee}$, ou seja, a proposição S pode ser escrita como $C \underline{\vee} D$.

Na valoração de uma disjunção exclusiva, o resultado será verdadeiro se, e somente se, apenas umas das proposições simples forem verdadeiras.

Conectivo condicional (Implicação lógica) – se... Então

Dadas as proposições simples A, B, elas formam uma condicional (ou implicação lógica) se for possível construir a estrutura: **se** A, **então** B (BISPO; CASTANHEIRA, 2011). A primeira proposição é chamada antecedente, e a segunda consequente. A condicional significa que a verdade da primeira proposição implica a verdade da segunda proposição (GERSTING, 2017). O símbolo usado para representar a implicação lógica é o $\rightarrow$, logo a regra se A, então B, pode ser escrita como $A \rightarrow B$.

Na valoração do condicional, se o antecedente e o consequente forem verdadeiros então o resultado será verdadeiro. Ou seja, $(V \to V = V)$. Porém, se o antecedente for verdadeiro e o consequente falso, o resultado será falso $(V \to F = F)$. Vejamos um exemplo:

A. O interruptor da sala foi desligado.
B. A luz da sala apagou.
C. $A \to B$

A proposição C deve ser traduzida como “Se o interruptor da sala for desligado, então a luz se apagará”.

Se as duas proposições realmente acontecerem, então temos o caso $V \to V = V$, ou seja, C é verdade. Porém, se o interruptor for desligado, mas por algum motivo a luz não se apagar, então temos o caso $V \to F = F$, ou seja, C é falso.

Conectivo Bicondicional – se, e somente se

Dadas as proposições simples A, B, elas formam uma bicondicional se for possível construir a estrutura: A **se, e somente se**, B (BISPO; CASTANHEIRA, 2011). O símbolo usado para representar esse conectivo é o ↔, então a expressão A **se, e somente se**, B, pode ser expressa simbolicamente por $A \leftrightarrow B$. Vejamos um exemplo. Considere as proposições a seguir:

P. Lucas receberá o dinheiro.
Q. Lucas completará o trabalho.
R. $A \leftrightarrow B$

A proposição R, deve ser traduzida como “Lucas receberá o dinheiro se, e somente se, completar o trabalho”.

A valoração do conectivo bicondicional será verdadeira quando o valor-lógico das duas proposições forem iguais, tanto para verdadeiro como para falso. Ou seja, $V \leftrightarrow V = V$ e $F \leftrightarrow F = V$.

Equivalência lógica

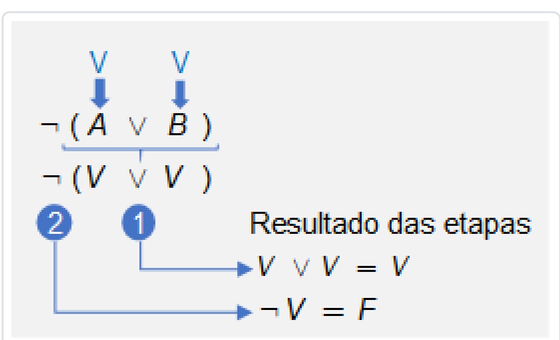
No conectivo bicondicional apresentamos uma definição trazida por Gersting (2017) que diz que a fórmula (i) $A \leftrightarrow B$ é um atalho para (ii) $(A \to B) \wedge (B \to A)$. A autora usa o termo “atalho” porque o resultado da fórmula (i) é igual ao da (ii) para todas as combinações possíveis de entradas, isso acontece porque estamos diante de uma **equivalência lógica**.

Além do conectivo bicondicional, existem outras importantes equivalências lógicas que serão estudadas em outra seção. Nesse momento, vamos apresentar outras duas equivalências lógicas, chamadas Leis de De Morgan que foram obtidas e demonstradas pelo matemático inglês Augustus De Morgan. As duas equivalências são:

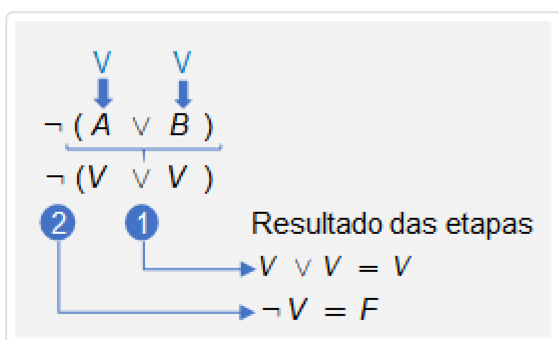
I. $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$
II. $\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$

Observe na fórmula I que, do lado esquerdo da equivalência, a negação está sendo aplicada ao resultado de uma disjunção, enquanto do lado direito a negação afeta cada uma das proposições. Para treinarmos, vamos verificar a veracidade da fórmula I. Para ser uma equivalência, o resultado precisa ser igual para todas as combinações possíveis de entrada. Então temos as seguintes combinações possíveis para as proposições A, , B: **(1)** A = V e B = V; **(2)** A = V e B = F; **(3)** A = F e B = V; **(4)** A = F e B = F. Vamos testar essas quatro combinações de entrada, primeiro para a fórmula $\neg(A \vee B)$. Será ilustrado cada passo para a valorização da fórmula, para cada combinação possível de entrada. Como resultado lógico temos que para as entradas (1), (2), e (3) o resultado é F, já para a entrada (4) o resultado é V.

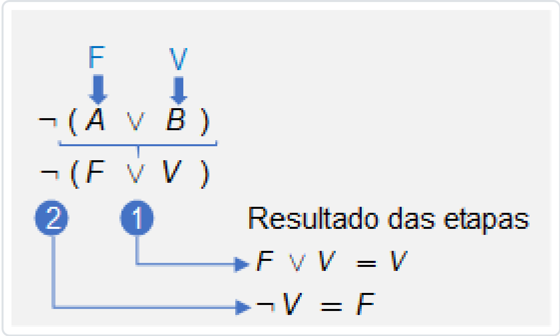
(1) A = V e B = V



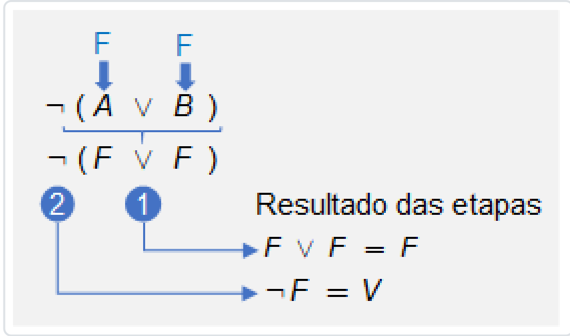
(2) A = V e B = F



(3) A = F e B = V



(4) A = F e B = F



Estudamos os principais conectivos lógicos. Para complementar seu conhecimento, consulte o livro didático e veja o desenvolvimento da valorização da fórmula, do lado direito, referente a equivalência lógica.

Lógica Computacional

Métodos dedutivos e inferência lógica

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nessa webaula será apresentado duas regras de equivalência para a Lógica Proposicional: a regra de equivalência e a regra de inferência

Regras de equivalência de dedução para a Lógica Proposicional

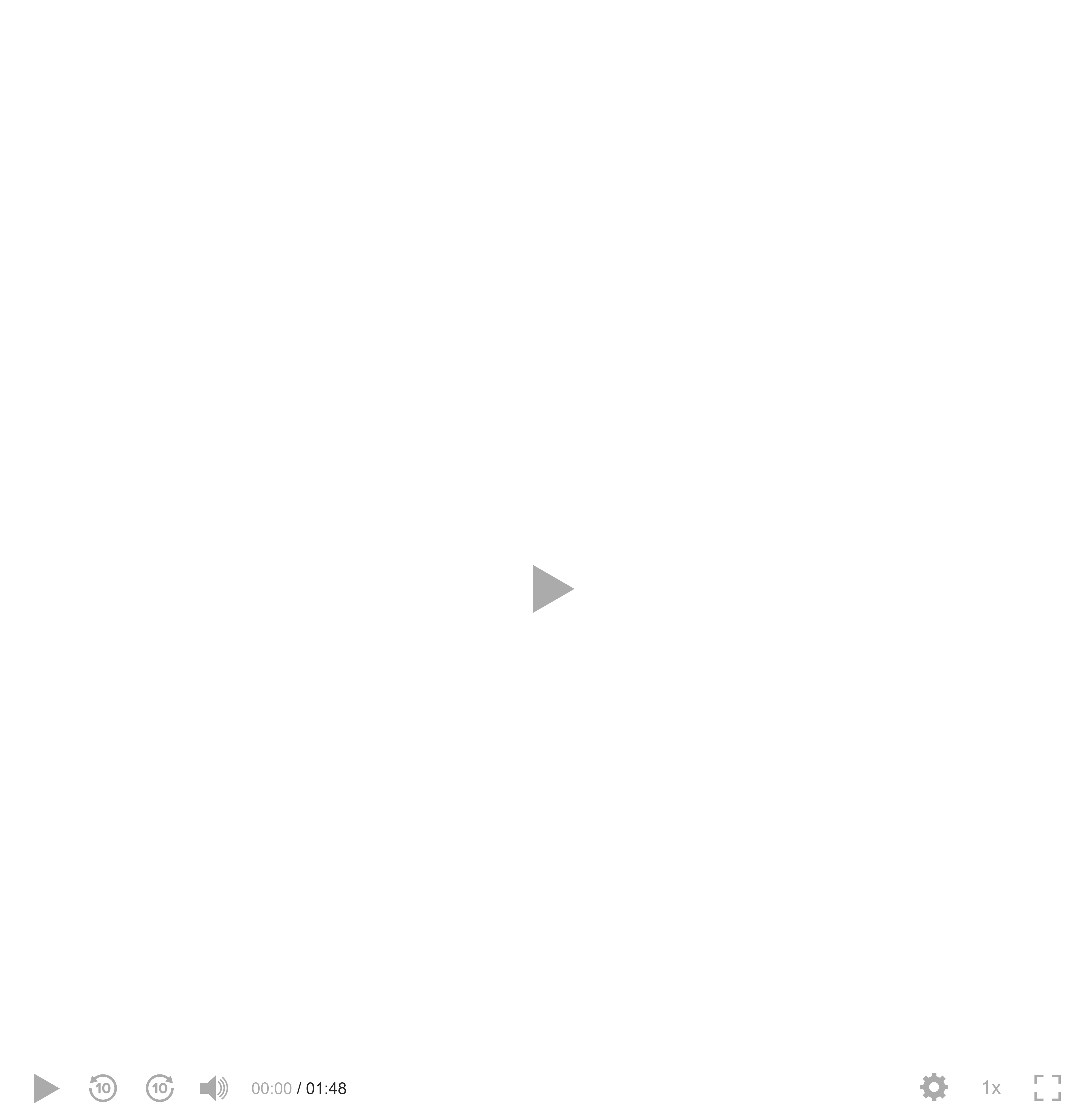
As regras de dedução são divididas em dois tipos: **regras de equivalência** e **regras de inferência**. Lembrando que duas fbfs são equivalentes, quando todas as combinações possíveis de entradas geram o mesmo resultado de saída para ambas as fbfs, as regras de equivalência serão usadas quando uma fbf (que pode ser uma hipótese ou resultado de uma regra) pode ser substituída por outra fbf, mantendo o resultado lógico. Por exemplo, se considerarmos a fbf que traduz uma das leis de De Morgan: $\neg\,(A \vee B) \Leftrightarrow \neg A \wedge \neg B$, em uma situação adequada podemos substituir a fbf $\neg(A \vee B)$ por $\neg A \wedge \neg B$, pois ambas são equivalentes. No quadro estão elencadas as regras de equivalência que iremos utilizar.

	Expressão (fbf)	Equivalente (fbf)	Nome/Abreviação
1.	$P \vee Q$	$Q \vee P$	Comutatividade/com
2.	$P \vee Q$	$Q \wedge P$	
3.	$(P \vee Q) \vee R$	$P \vee (Q \vee R)$	Associatividade/ass
4.	$(P \wedge Q) \wedge R$	$P \wedge (Q \wedge R)$	
5.	$\neg(P \vee Q)$	$\neg P \wedge \neg Q$	Leis de De Morgan/De Morgan
6.	$\neg(P \wedge Q)$	$\neg P \vee \neg Q$	
7.	$P \rightarrow Q$	$\neg P \vee Q$	Condicional/cond
8.	P	$\neg(\neg P)$	Dupla negação/dn
9.	$P \leftrightarrow Q$	$(P \rightarrow Q) \wedge (Q \rightarrow P)$	Definição de equivalência/que

No quadro você pode verificar que existem seis conjuntos de regras de dedução, sua utilização será da seguinte forma: Se tivermos uma expressão como da linha 1, $P \vee Q$, quando necessário, podemos substitui-la por $Q \vee P$, pois essas fbfs são equivalentes e trata-se da propriedade da comutatividade. O contrário também é válido, quando aparecer $Q \vee P$, podemos substituir por $P \vee Q$. Esse processo de substituir uma fbf por outra, é o mesmo para todas as demais regras apresentadas.

Para finalizar essa webaula é importante ter em mente que nas regras de equivalência, as colunas podem ser usadas nos dois sentidos. Já nas regras de inferência, só existe um sentido: a fbf da coluna “De” pode ser substituída pela coluna de “Podemos deduzir”, mas o contrário não é verdade. Continue estudando!

Agora, clique aqui e acesse ao jogo LOGIKABOOM e treine seus conhecimentos!



Para visualizar o vídeo, acesse seu material digital.