1° passo:

é escrever as premissas e conclusão dentro de uma lista (lembrando que na frente da conclusão sempre vai um '¬' na frente

EX: (P ⊃ Q), P ∴ Q

Vai ficar [(P ⊃ Q), P, ¬Q]

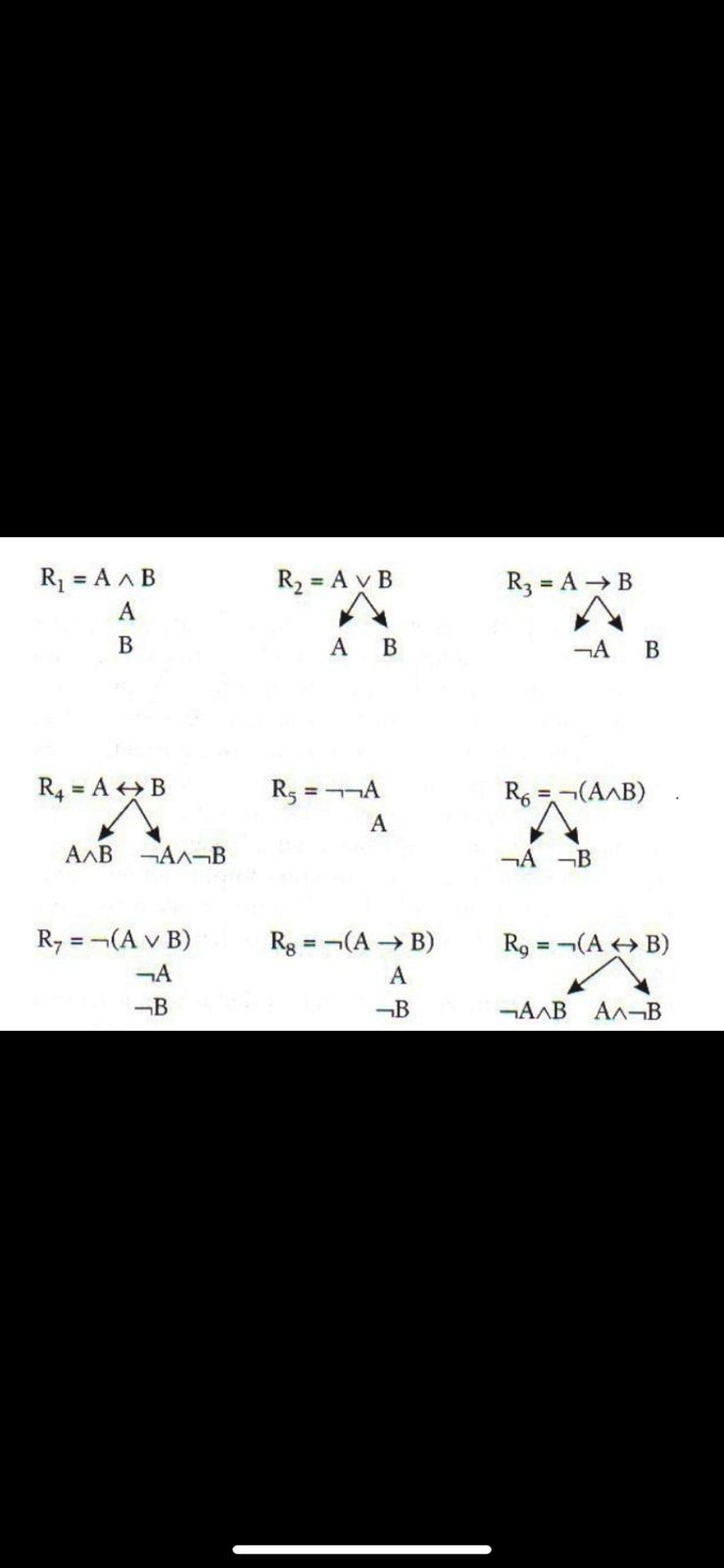
2° passo:

Depois disso precisamos pegar todos as fórmulas que não estão no seu estado atômico e simplificar (Já vou mandar a foto de como fazer isso). Depois disso coloca o resultado em nova lista e apaga a fórmula não simplificada da lista original

EX: [(P ⊃ Q), P, ¬Q] (1) vira [P, ¬Q] (1)

e a outra lista fica assim [¬P, Q] (2)

(Isso porque (P ⊃ Q) simplificado vira ¬P e Q)



3° passo:

Agora a gente vai misturar a lista (1) e a lista (2). E fazer uma lista de listas, dessa forma a seguir:

[[P, ¬Q, ¬P], [P, ¬Q, Q]]

4° passo:

Agora dentro dessa lista a gente vai validar se existe alguma contradição, e se existe a gente apaga a lista (seria o equivalente a um ramo fechado do tableaux)

No caso acima as duas listas contêm contradições então apagamos ambas

EX: [[], []] ou []

5° passo:

Depois a gente cria um código que retorna True quando a lista for vazia (ou seja todos os ramos foram fechados) e False quando a lista não estiver vazia (ou seja algum ramo não foi fechado)

Em outras palavras, se retornar False (lista não vazia) significa que o argumento é invalido

Se retornar True (lista vazia) significa que o argumento é válido