

int.1 Fórmulas

Eis a abordagem que usaremos para especificar rigorosamente as **sentenças** da lógica de primeira ordem e para lidar com as questões decorrentes do uso de **variáveis**. Primeiro definimos um conjunto *diferente* de expressões: **fórmulas**. Uma vez feito isso, podemos considerar o papel que as **variáveis** desempenham nelas—e, com base em algumas outras ideias, nomeadamente as de **variáveis** “livres” e “ligadas”, podemos definir o que é uma **sentença** (nomeadamente, uma **fórmula** sem **variáveis** livres). Fazemos isso não apenas porque torna a definição de “**sentença**” mais gerenciável, mas também porque será crucial para a maneira como definimos a noção semântica de satisfação.

Vamos definir “**fórmula**” para uma linguagem simples de primeira ordem, contendo apenas um único **símbolo de predicado** P e um único **símbolo de constante** a , e apenas os símbolos lógicos $\neg$, $\wedge$ e $\exists$. Nossas definições completas serão muito mais gerais: permitiremos infinitos **símbolos de predicado** e **símbolos de constante**. De fato, também consideraremos **símbolos de função** que podem ser combinados com **símbolos de constante** e **variáveis** para formar “termos”. Por enquanto, a e as variáveis serão nossos únicos termos. Precisamos de infinitas **variáveis**. Oficialmente usaremos os símbolos $v_0, v_1, \dots$, como variáveis.

Definição int.1. O conjunto de **fórmulas** Frm é definido da seguinte forma:

- fol:int:fml: 1. $P(a)$ e $P(v_i)$ são **fórmulas** ($i \in \mathbb{N}$).
fmls-atom
- fol:int:fml: 2. Se φ é uma **fórmula**, então $\neg\varphi$ é uma **fórmula**.
fmls-not
- fol:int:fml: 3. Se φ e ψ são **fórmulas**, então $(\varphi \wedge \psi)$ é uma **fórmula**.
- fol:int:fml: 4. Se φ é uma **fórmula** e x é uma **variável**, então $\exists x \varphi$ é uma **fórmula**.
fmls-ex
- fol:int:fml: 5. Nada mais é uma **fórmula**.
fmls-limit

(1) nos diz que $P(a)$ e $P(v_i)$ são **fórmulas**, para qualquer $i \in \mathbb{N}$. Estas são as chamadas **fórmulas atômicas**. Elas nos dão algo com que começar. As outras cláusulas nos dão maneiras de formar novas **fórmulas** a partir daquelas que já formamos. Assim, por exemplo, por (2), obtemos que $\neg P(v_2)$ é uma **fórmula**, já que $P(v_2)$ já é uma **fórmula** por (1). Então, por (4), obtemos que $\exists v_2 \neg P(v_2)$ é outra **fórmula**, e assim por diante. (5) nos diz que *somente* cadeias que podemos formar desta maneira contam como **fórmulas**. Em particular, $\exists v_0 P(a)$ e $\exists v_0 \exists v_0 P(a)$ *contam* como **fórmulas**, e $(\neg P(a))$ não conta, por causa dos parênteses externos supérfluos.

Esta maneira de definir **fórmulas** é chamada de *definição indutiva*, e nos permite provar coisas sobre **fórmulas** usando uma versão de prova por indução chamada *indução estrutural*. Estas são discutidas de maneira geral em ?? e ??, que você deve revisar antes de mergulhar nas provas mais adiante. Basicamente, a ideia é que, se você quer dar uma prova de que algo é verdadeiro para todas as **fórmulas**, mostra primeiro que é verdadeiro para as **fórmulas** atômicas e então que *se* for verdadeiro para qualquer **fórmula** φ (e ψ), *também*

é verdadeiro para $\neg\varphi$, $(\varphi \wedge \psi)$ e $\exists x \varphi$. Por exemplo, isso prova que é verdadeiro para $\exists v_2 \neg P(v_2)$: da primeira parte você sabe que é verdadeiro para a fórmula atômica $P(v_2)$. Então obtém que é verdadeiro para $\neg P(v_2)$ pela segunda parte, e novamente que é verdadeiro para $\exists v_2 \neg P(v_2)$ em si. Como todas as fórmulas são geradas indutivamente a partir de fórmulas atômicas, isso funciona para qualquer uma delas.

Créditos das Imagens

Referências Bibliográficas