The two typing relations

Question: What is the relation between these two statements?

1. $\texttt{t : T}$
2. $\vdash \texttt{t : T}$

The two typing relations

Question: What is the relation between these two statements?

1. $\mathtt{t} : \mathtt{T}$
2. $\vdash \mathtt{t} : \mathtt{T}$

First answer: These two relations are completely different things.

- We are dealing with several different small programming languages, *each with its own typing relation* (between terms in that language and types in that language)
- For the simple language of numbers and booleans, typing is a *binary* relation between terms and types ($\mathtt{t} : \mathtt{T}$).
- For $\lambda_{\rightarrow}$, typing is a *ternary* relation between contexts, terms, and types ($\Gamma \vdash \mathtt{t} : \mathtt{T}$).

  (When the context is empty — because the term has no free variables — we often write $\vdash \mathtt{t} : \mathtt{T}$ to mean $\emptyset \vdash \mathtt{t} : \mathtt{T}$.)

Conservative extension

Second answer: The typing relation for $\lambda_{\rightarrow}$ *conservatively extends* the one for the simple language of numbers and booleans.

- Write "language 1" for the language of numbers and booleans and "language 2" for the simply typed lambda-calculus with base types `Nat` and `Bool`.
- The terms of language 2 include all the terms of language 1; similarly typing rules.
- Write $\mathtt{t} :_1 \mathtt{T}$ for the typing relation of language 1.
- Write $\Gamma \vdash \mathtt{t} :_2 \mathtt{T}$ for the typing relation of language 2.
- *Theorem:* Language 2 conservatively extends language 1: If `t` is a term of language 1 (involving only booleans, conditions, numbers, and numeric operators) and `T` is a type of language 1 (either `Bool` or `Nat`), then $\mathtt{t} :_1 \mathtt{T}$ iff $\emptyset \vdash \mathtt{t} :_2 \mathtt{T}$.