$$\frac{\begin{array}{c} t_1 \in \textbf{UserTable} \\ (t_1, t_2 :: c_2) \in \textbf{OwnerModel} \vee (t_1, -, t_2, -) \in \textbf{MemberModel} \\ (t_2, t_3 :: c_3) \in (\textbf{OOR} \cup \textbf{ONR}) \end{array}}{(t_1, t_3 :: c_3) \in \textbf{HierarchyModel}} \quad \text{[Hi-Model-Init]}$$

$$\frac{\begin{array}{c} (t_1, t_2 :: c_2) \in \textbf{HierarchyModel} \\ (t_2, t_3 :: c_3) \in (\textbf{OOR} \cup \textbf{ONR}) \end{array}}{(t_1, t_3 :: c_3) \in \textbf{HierarchyModel}} \quad \text{[Hi-Model-Tran]}$$

Figure 13: Rules for hierarchical model inference.

$$\frac{\begin{array}{c} (t_1, t_3 :: c_3) \in \textbf{HierarchyModel} \quad l^C : \langle -, t_3, -, - \rangle \\ (t_1, t_3 :: c_3, C) \in \textbf{SafeC} \end{array}}{l^C \in \textbf{SafeOp}} \quad \text{[Hi-Check]}$$

$$\frac{\begin{array}{c} (t_1, t_3 :: c_3) \in \textbf{HierarchyModel} \\ (t_1, t_2 :: c_2) \in \textbf{HierarchyModel} \quad (t_2, t_3 :: c_3) \in (\textbf{OOR} \cup \textbf{ONR}) \\ (t_2 :: c_p, t_3 :: c_3) \in C \\ (t_1, t_2 :: c_2, C) \in \textbf{SafeC} \end{array}}{(t_1, t_3 :: c_3, C) \in \textbf{SafeC}} \quad \text{[Hi-Check-Recursive]}$$

$$\frac{\begin{array}{c} (t_1, t_3 :: c_3) \in \textbf{HierarchyModel} \\ (t_1, t_2 :: c_2) \in \textbf{OwnerModel} \quad (t_2, t_3 :: c_3) \in (\textbf{OOR} \cup \textbf{ONR}) \\ (t_2 :: c_p, t_3 :: c_3) \in C \\ (t_1 :: c_p, t_2 :: c_2) \in C \end{array}}{(t_1, t_3 :: c_3, C) \in \textbf{SafeC}} \quad \text{[Hi-Check-Owner]}$$

$$\frac{\begin{array}{c} (t_1, t_3 :: c_3) \in \textbf{HierarchyModel} \\ (t_1, t_4 :: c_4, t_2, t_4' :: c_4') \in \textbf{MemberModel} \quad (t_2, t_3 :: c_3) \in (\textbf{OOR} \cup \textbf{ONR}) \\ (t_2 :: c_p, t_3 :: c_3) \in C \\ (t_1 :: c_p, t_4 :: c_4) \in C \quad (t_2 :: c_p, t_4' :: c_4') \in C \end{array}}{(t_1, t_3 :: c_3, C) \in \textbf{SafeC}} \quad \text{[Hi-Check-Member]}$$

Figure 14: Rules for missing hierarchical check.

A INFERRING AND CHECKING RULES FOR HIERARCHICAL MODELS

Figure 13 presents the rules for inferring hierarchical models. According to its definition (Section 3), we initially use [Hi-Model-Init] to build hierarchical models based on existing ownership or membership models. The model between table `user` and table `notice` (Figure 2) is such a case. Next, hierarchical models are transitively deduced via 1:1 and 1:n relationship, as demonstrated by [Hi-Model-Tran].

Figure 14 illustrates the principle of detecting missing hierarchical checks. The set **SafeC** is a set of triples like $(t_1, t_3 :: c_3, C)$, indicating that C properly checks the hierarchical model between t_1 and $t_3 :: c_3$. Rule [Hi-Check] is intuitive, reducing the safety check of sensitive operation l^C to verifying the existence of $(t_1, t_3 :: c_3, C) \in \textbf{SafeC}$. The three remaining rules "inductively" verify a given hierarchical model:

- Basis: Rule [Hi-Check-Owner] and [Hi-Check-Member] handle a root hierarchical model based on ownership and membership models, respectively.
- Induction: Rule [Hi-Check-Recursive] validates the hierarchical model layer by layer. Interestingly, this rule operates in the opposite direction of [Hi-Model-Tran].