T-DPLL: Online Schema

Input:
Original T-formula $\varphi$
Reference to a T-Assignment $\mu$ (a set of T-literals) which is initially empty

T-preprocess:
Simplifies $\varphi$ into a simpler version ($\varphi^P$) using
- Boolean preprocessing from DPLL
- Theory-dependent rewriting steps on T-literals of $\varphi$
Updates $\mu$ if necessary
Returns UNSAT if a conflict is encountered

T-decide-next-branch
Selects literal to branch on as in DPLL

T-deduce
Iteratively deduces the implied Boolean literals and updates $\varphi^p$ and $\mu^p$ until:
1. $\mu^p$ violates $\varphi^p$ (and entails $\{\} \}$) – return Conflict
2. $\mu^p$ satisfies $\varphi^p$, and the result is run through the T-solver
   - satisfied by T-solver? return Sat
   - unsatisfied by T-solver? return Conflict
3. Nothing new can be deduced – return Unknown

T-analyze-conflict
Determine the Boolean conflict set, $n^p$; add it to $\varphi^p$, then backtrack up to blevel