Propositional Validity Checker

Task. Implement a validity checker of propositional formulas in form of $\varphi \to \psi$, where φ and ψ use $\wedge$, $\vee$, $\neg$, and propositional variables.

Background.

- We can evaluate a formula over variables $x_1, x_2, \dots x_n$ given an assignment $\{x_1 \mapsto v_1, x_2 \mapsto v_2, \dots\}$.
- We say that formula is unsatisfiable if it evaluates to $false$ under all possible assignments.
  - We may try all the assignments one by one to determine satisfiability.
- We say that formula is valid if its negation is unsatisfiable.
  - Therefore, $\varphi \to \psi$ is valid when $\varphi \wedge \neg\psi$ is unsatisfiable, otherwise it is invalid.
  - If the negation evaluates to $true$, the assignment is called a *counterexample* to the validy query.

Input.

- The checker is given a filename as its only command-line parameter.

```
$ java ValidityChecker definition.txt
```

- The file contains one line. An example of the syntax of the input file `definition.txt`:

```
AND(X, OR(Y, Z)) => AND(X, Y)
```

Output.

- **Valid.** or **Invalid. (Counterexample: [X** $= true$**, Y** $= false$**, . . .])**