Transformation Tool Contest:

Feature Diagrams to Dot

Vadim Zaytsev, UTwente, NL

13 June 2025, TTC @ STAF 2025

UNIVERSITY OF TWENTE.

Formal Methods & Tools

Solution Pragmatics

- software language engineering
  - “language oriented programming”
- F# as the platform
- as concise as possible
  - “grammar” is 1 LOC, 4 words
  - “transformation” is 6 LOC
  - +200 LOC of F# code
- integration was a challenge

Use of LLMs

- “use Python”
- too complex to explain
- exploratory programming
  - does not work
- *vibing* requires time!
- hitting the right level of abstraction
- some minor but annoying bugs in code
- cf. OOPSLE keynote of Jordi Cabot

The Solution

- the input is indentation based
  - Feature ::= (**root** ⇒ [**make** ⇒ **goal**])
- **root** might not be necessary…
- *nothing* needs to happen to constraints! ⇒ semi-parsing
- transformaion is less thought through
  - **draw** FeatureModel
  - **template** feature ::= 'A → B [arrowhead="HEAD" arrowtail="TAIL" dir="both"]'
  - **each** mandatory ⇒ feature **with** HEAD=dot TAIL=none
  - **each** optional ⇒ feature **with** HEAD=odot TAIL=none
  - **each** alternative ⇒ feature **with** HEAD=none TAIL=odot
  - **each** or ⇒ feature **with** HEAD=none TAIL=dot

Preliminary Performance

- **Initialization**
  - 4618500
  - vs
  - 81297000

×18

- **Load**
  - 7845083
  - vs
  - 1322074500

×170

- **LOC**
  - code
    - 8453 C# vs 205+ F#
  - grammar
    - 58 AnyText vs 1 ???
  - transformation
    - (hardcoded) vs 6 ???