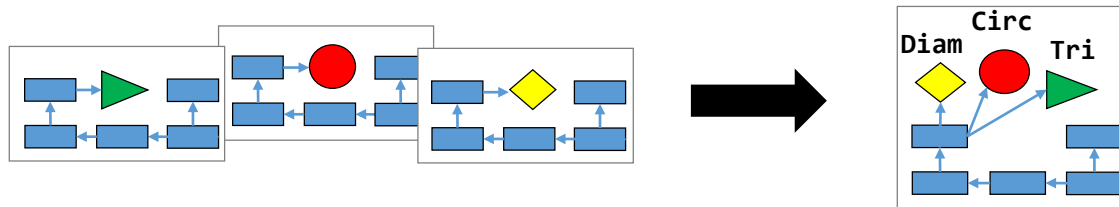# Three scenarios for learning GTs

Scenario 1: rule merging
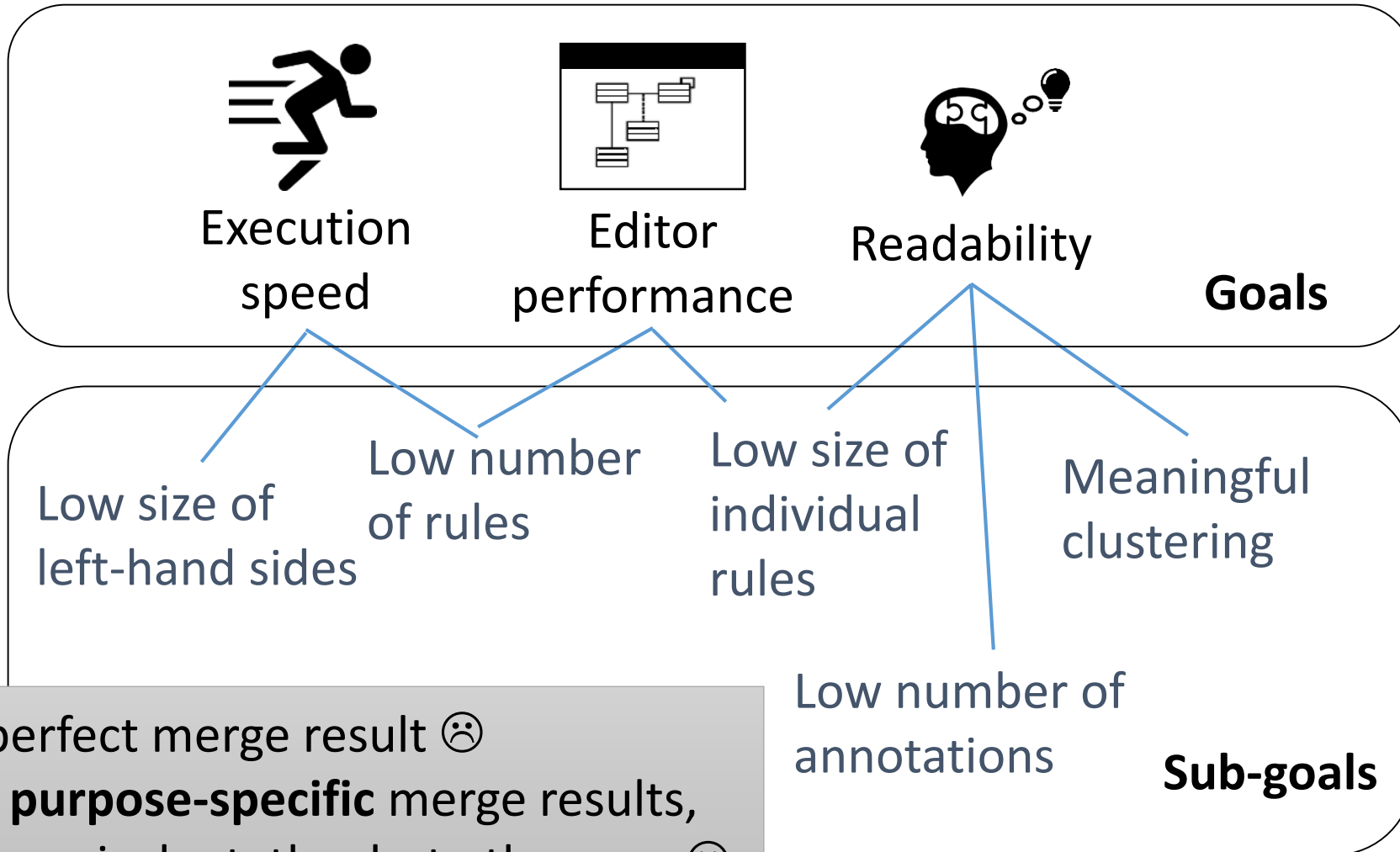

- Merge similar rule variants to produce a VB rule ("150% rule")
- Enable compact specification and improved performance

Relevant paper:
Daniel Strüber, Julia Rubin, Thorsten Arendt, Marsha Chechik, Gabriele Taentzer, Jennifer Plöger: Variability-based model transformation: formal foundation and application. Formal Aspects Comput. 30(1): 133-162 (2018)

Quality goals during merging may be in conflict

Execution speed

Editor performance

Readability

Goals

Low size of left-hand sides

Low number of rules

Low size of individual rules

Meaningful clustering

Low number of annotations

Sub-goals

- No **single** perfect merge result ☹
- Can derive **purpose-specific** merge results, all of them equivalent, thanks to theorem ☺

Scenario 2: mutation operator generation

- **Context**: When aiming to find an optimal model, the design space is usually too large for enumerating all possible solution models

- **Solution**: Use a genetic algorithm; guided search over fitness criteria. Quality of operators determines quality of the algorithm

Genetic algorithm
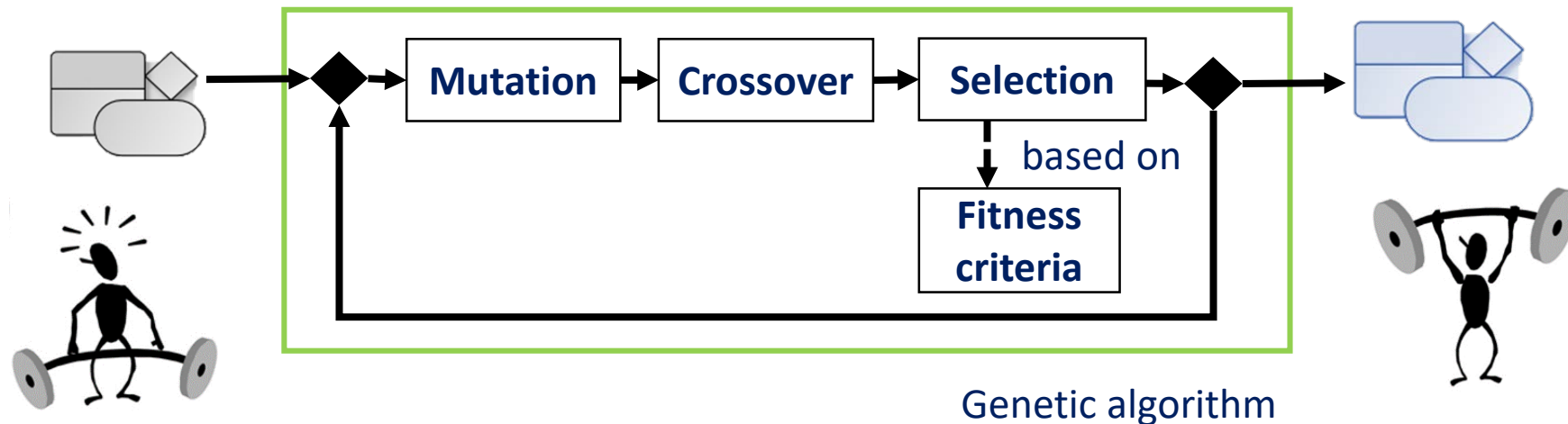
Relevant paper:
Daniel Strüber: Generating Efficient Mutation Operators for Search-Based Model-Driven Engineering. ICMT 2017: 121-137

Previous approach for mutation operator generation (ICMT'17)
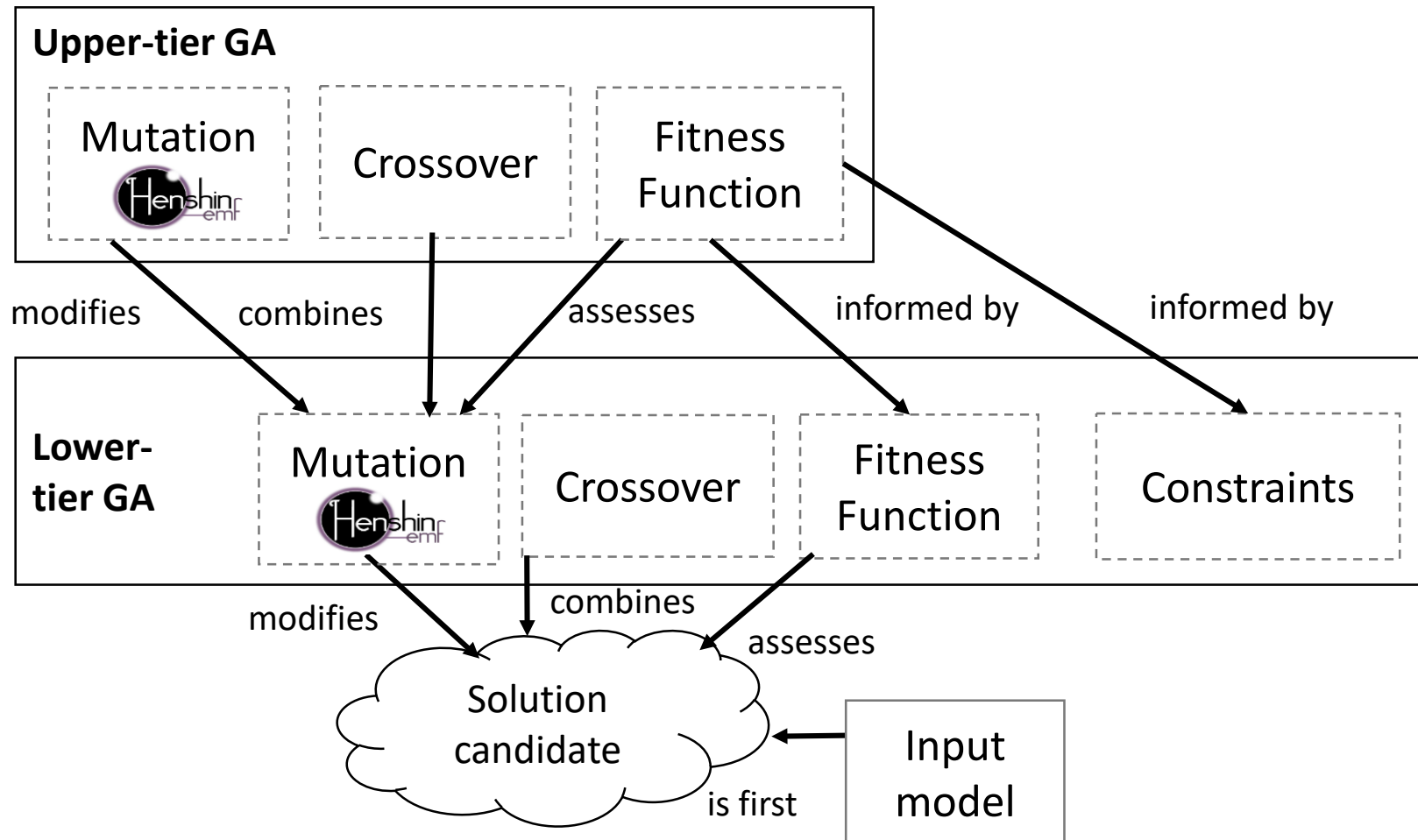
Scenario 3: transformation by example

Scenario 3: transformation by example