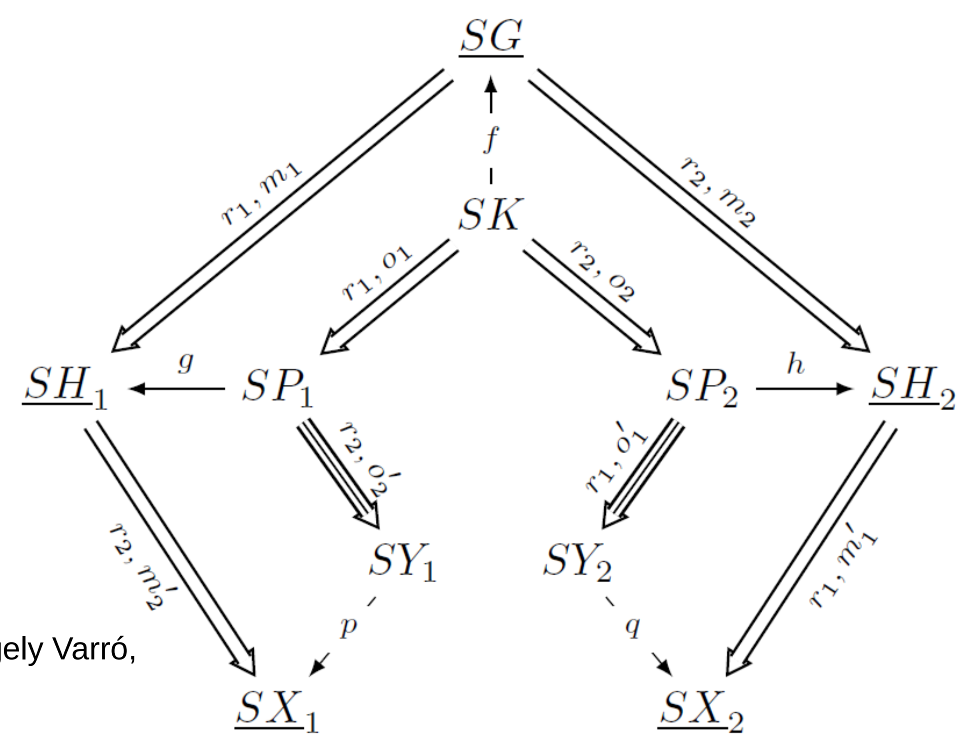
Improved Conflict Detection for Graph Transformation with Attributes

GaM 2015

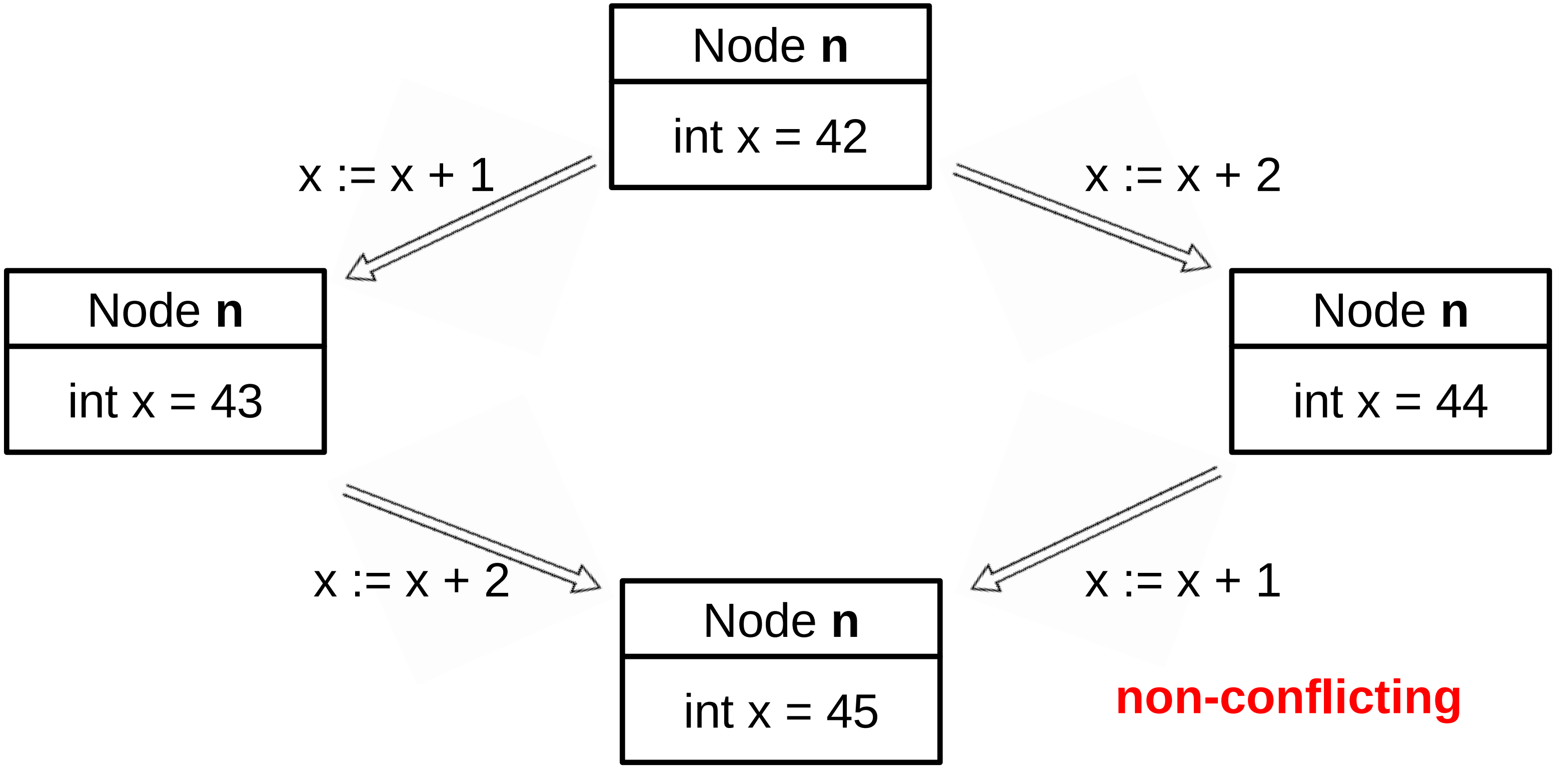


Géza Kulcsár, Frederik Deckwerth, Malte Lochau, Gergely Varró, Andy Schürr
geza.kulcsar@es.tu-darmstadt.de

Motivation: Conflict of Attribute Operations

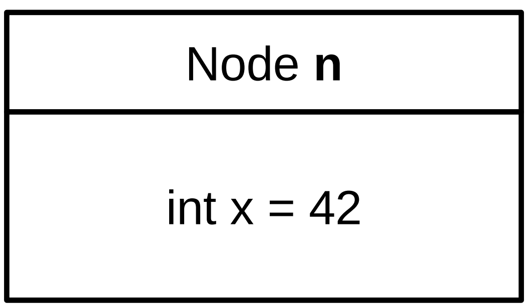
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Conflict:** having two alternative operations on the same object, applying the second after the first leads to a different result than applying the first after the second (or at least one sequence is not possible)

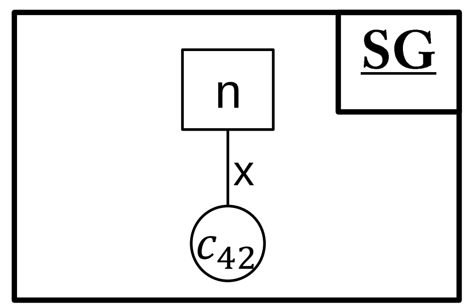


Representing Attributes: Symbolic Graphs

- **Symbolic graphs (Orejas):** graphs enriched with: (i) attribute nodes holding variables, (ii) attribute edges and (iii) a first-order logic formula to assign values to attribute variables
  - note that a symbolic graph can specify multiple models (attribute value assignments)



model

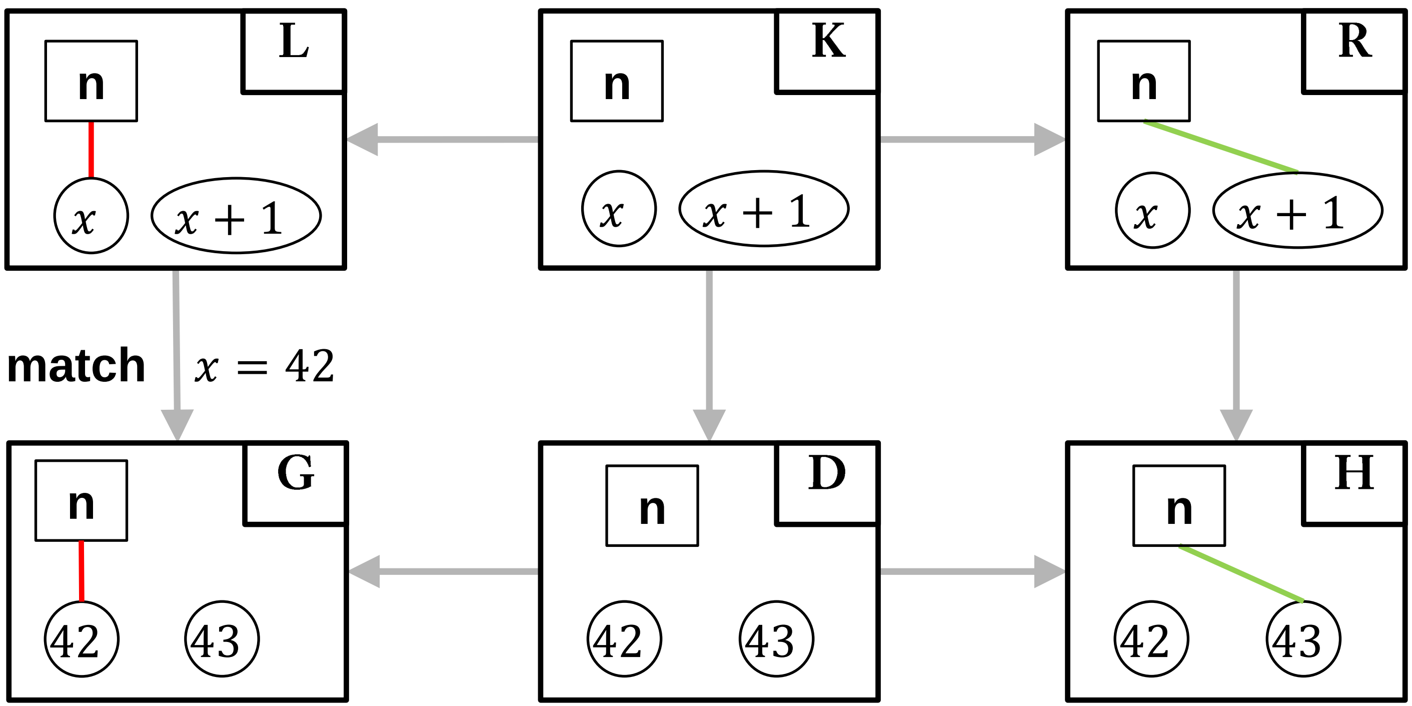


$\Phi: \{c_{42} = 42\}$

symbolic graph representation

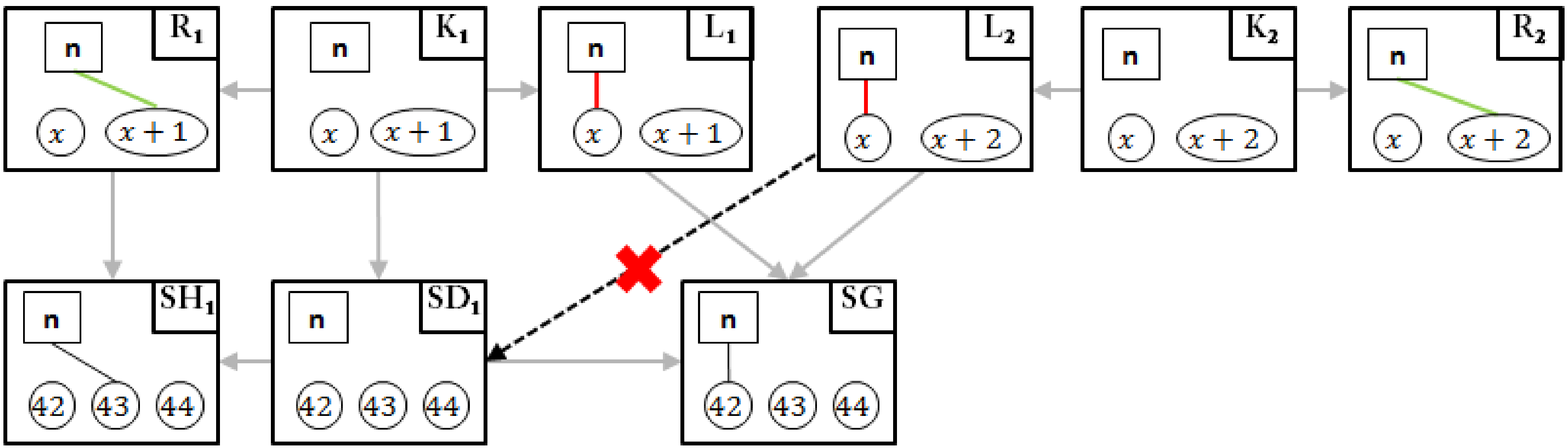
Graph Transformation with Attributes

- Graph transformation: a rule-based approach to modify graph-based models
- **Example rule:** attribute value is increased by 1



Example: Parallel Dependence and Attributes

Parallel independence of two direct derivations: no direct derivation deletes any element which is matched by the other direct derivation



- **...but still, 42 + 1 + 2 = 42 + 2 + 1**

Paralel Independence is Too Strict

TECHNISCHE UNIVERSITÄT DARMSTADT

- Parallel independence is intuitively too strict for analysing attribute operations (resulting in false positives, i.e., situations where the derivation sequences are equivalent may be recognized as conflict)

- **A new, more precise conflict condition is needed to take the semantics of attribute operations into account**

Contribution: an Improved Conflict Notion for Graphs with Attributes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Not independent of the classical approach based on parallel dependence

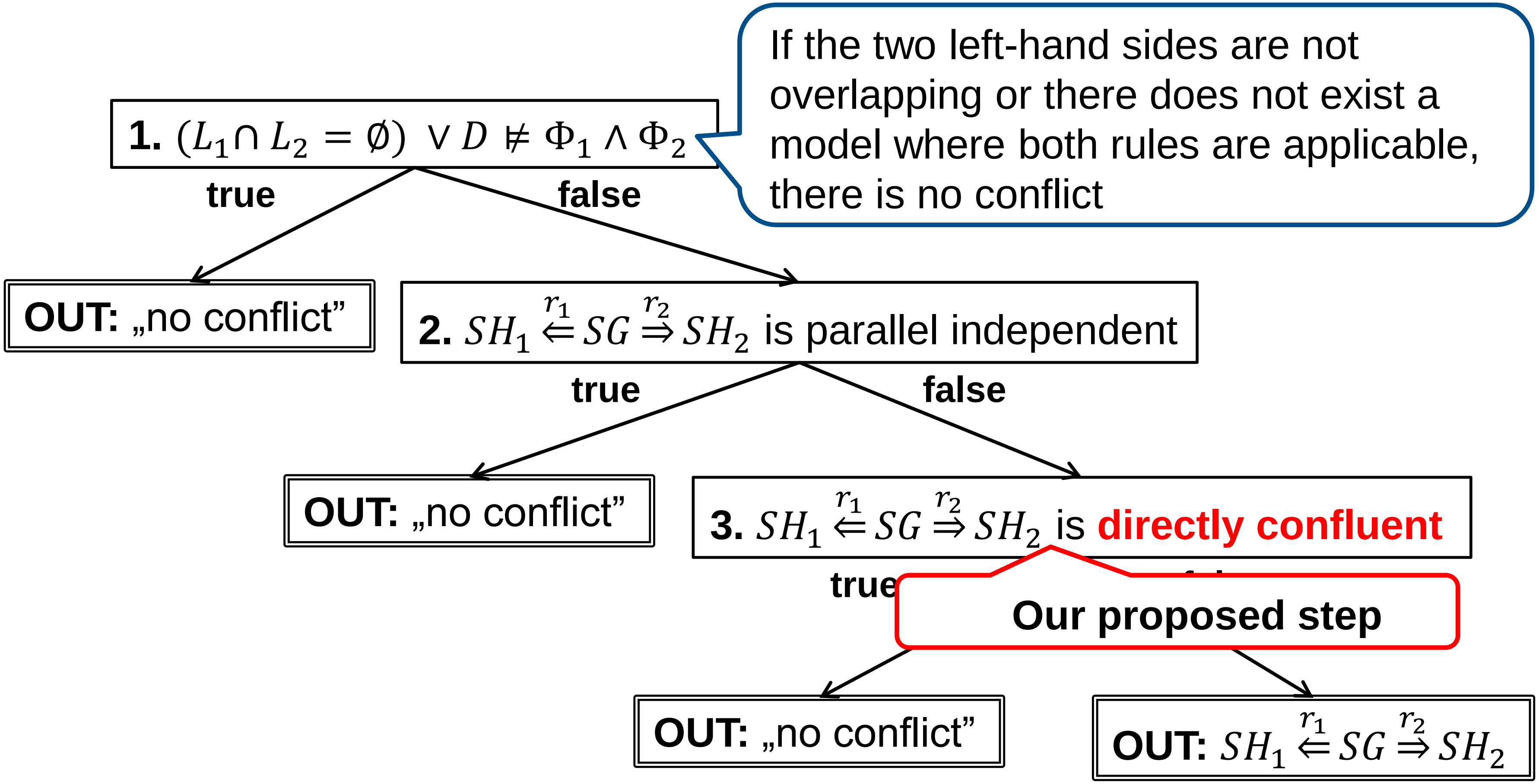
- Performed as a 3-step process (for a given pair of alternative direct derivations)

  1. Checking the two rules if there is any chance of a conflict
  2. Checking the parallel dependence of the direct derivations

  classical approach

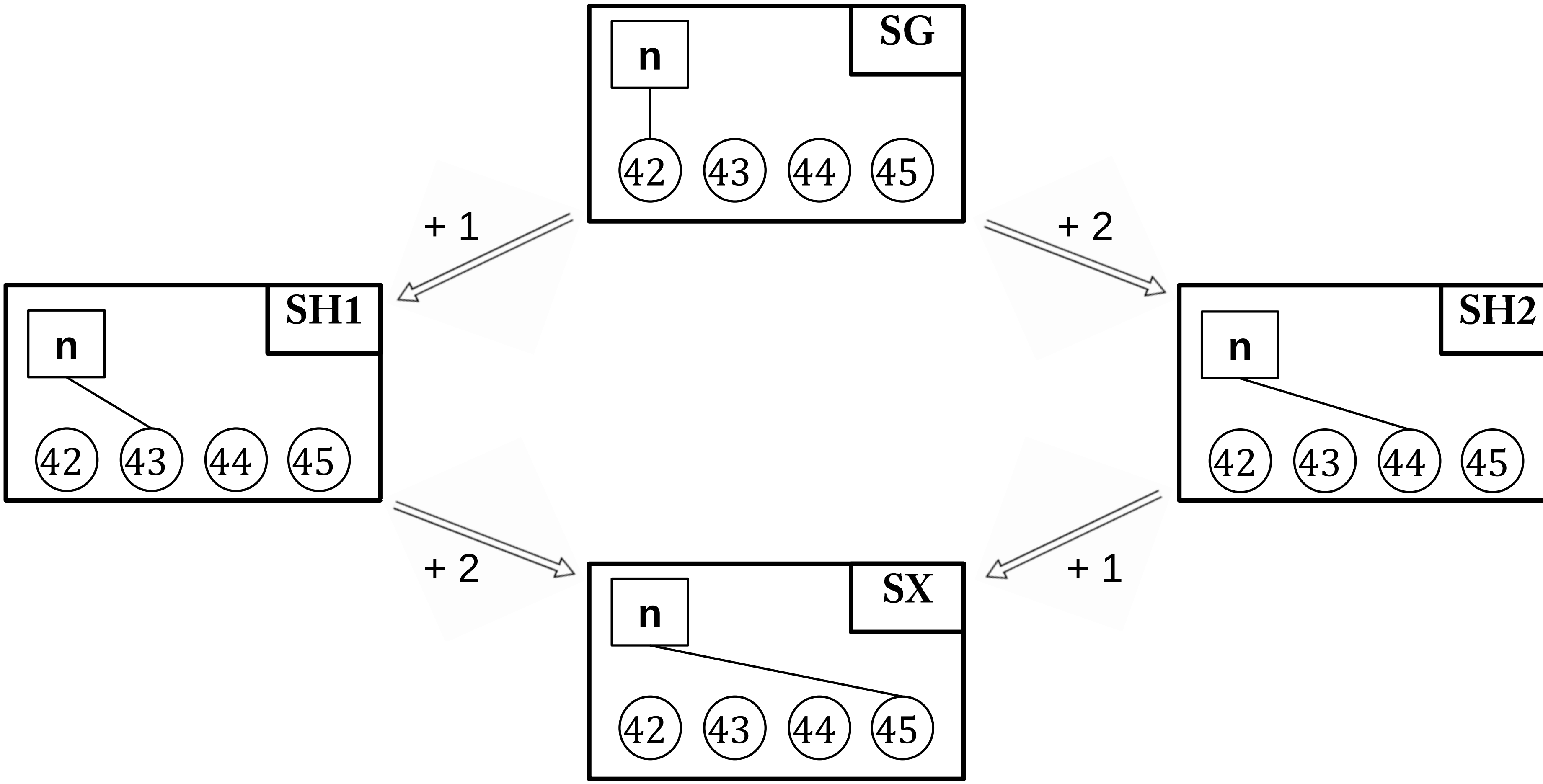
  3. Refining the conflict detection by checking **direct confluence** in case of parallel dependence **(contribution)**

A Conflict Detection Process



Direct Confluence

- ***Direct confluence*** is proposed as a less conservative conflict condition



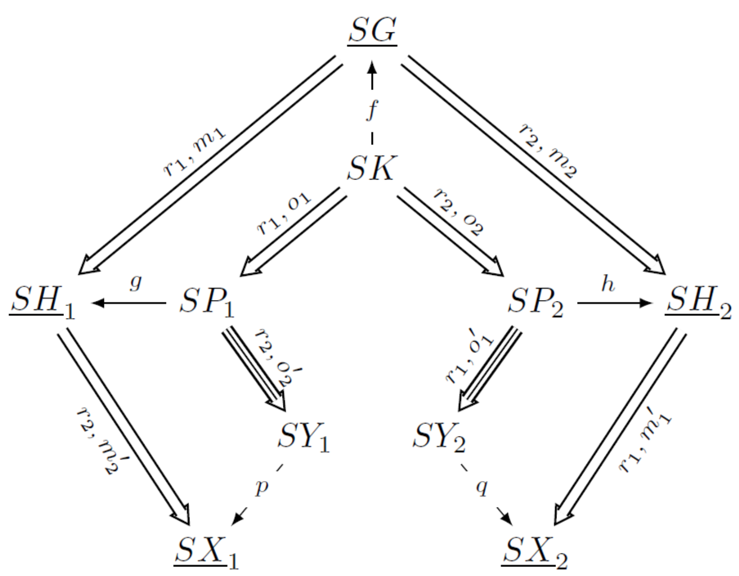
Direct Confluence

$$\begin{tikzcd}
& SG \arrow[dl, Rightarrow, "{r_1, m_1}"'] \arrow[dr, Rightarrow, "{r_2, m_2}"] & \\
SH_1 \arrow[dr, Rightarrow, "{r_2, m'_2}"'] & & SH_2 \arrow[dl, Rightarrow, "{r_1, m'_1}"] \\
& SX_1 \simeq SX_2 &
\end{tikzcd}$$

Result: Completeness

- Our approach has the *completeness* property:

Although direct confluence is less restrictive than parallel independence, our approach is still able to detect each possible conflict situation (according to the direct confluence condition).



Conclusion

- Conflict detection is important for real-life rule-based applications, e.g., program refactorings, visual languages, ...
- Our approach takes the semantics of attribute operations into account, which can potentially reduce the number of false positives while retaining completeness
- It can be used as a static analysis technique (by lifting the detection process to rule level using minimal contexts)
- Moreover, by using symbolic graphs, an implementation of the approach can use any off-the-shelf SMT solver for the formula part (ongoing work)