

BX Tutorial: Graph Transformation - Triple Graph Grammars -

BANFF Bidirectional Transformations
02-DECEMBER-2013



Banff International Research Station
for Mathematical Innovation and Discovery

Frank Hermann, frank.hermann@uni.lu

SECAN-Lab

Interdisciplinary Centre for Security, Reliability and Trust
Université du Luxembourg

OVERVIEW

Basic terminology (<10 min.)

- Key idea and general approach

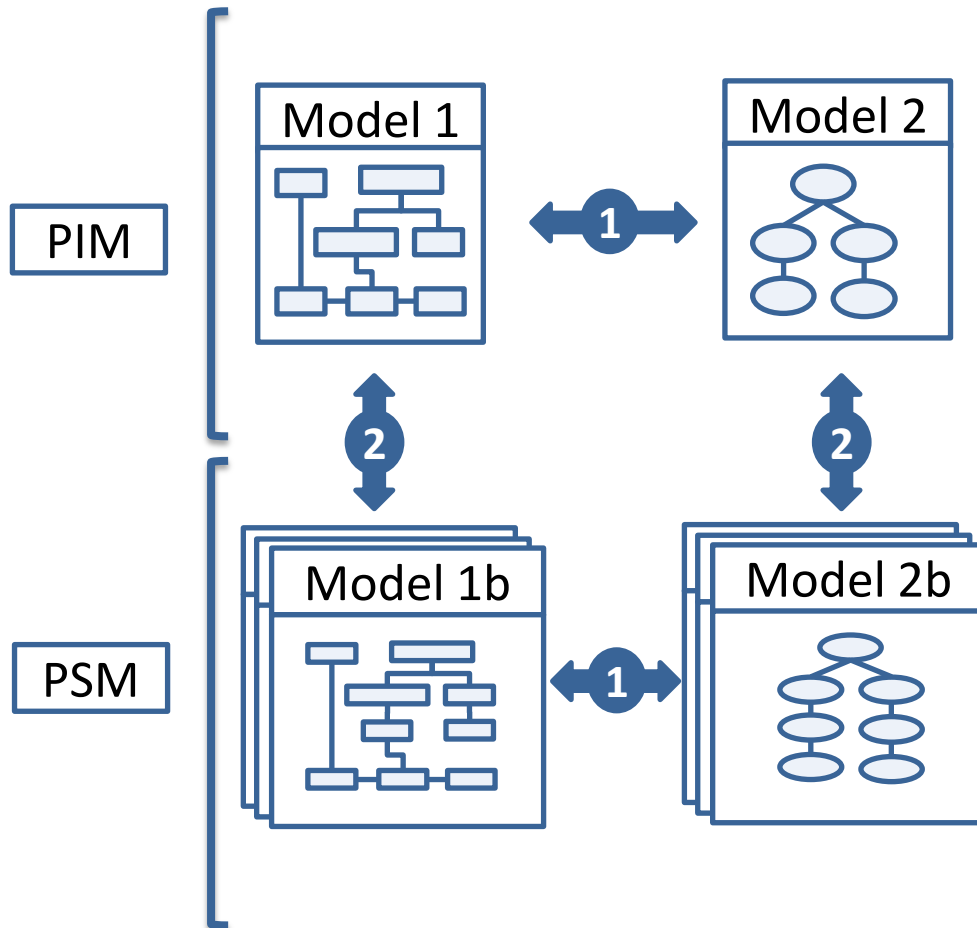
Core tutorial part (<25 min.)

- BX operations and analysis

Integration part (<10 min.)

- General concept for concurrent model synchronization
- Cross-disciplinary concepts

Interrelated Models in Model Driven Engineering

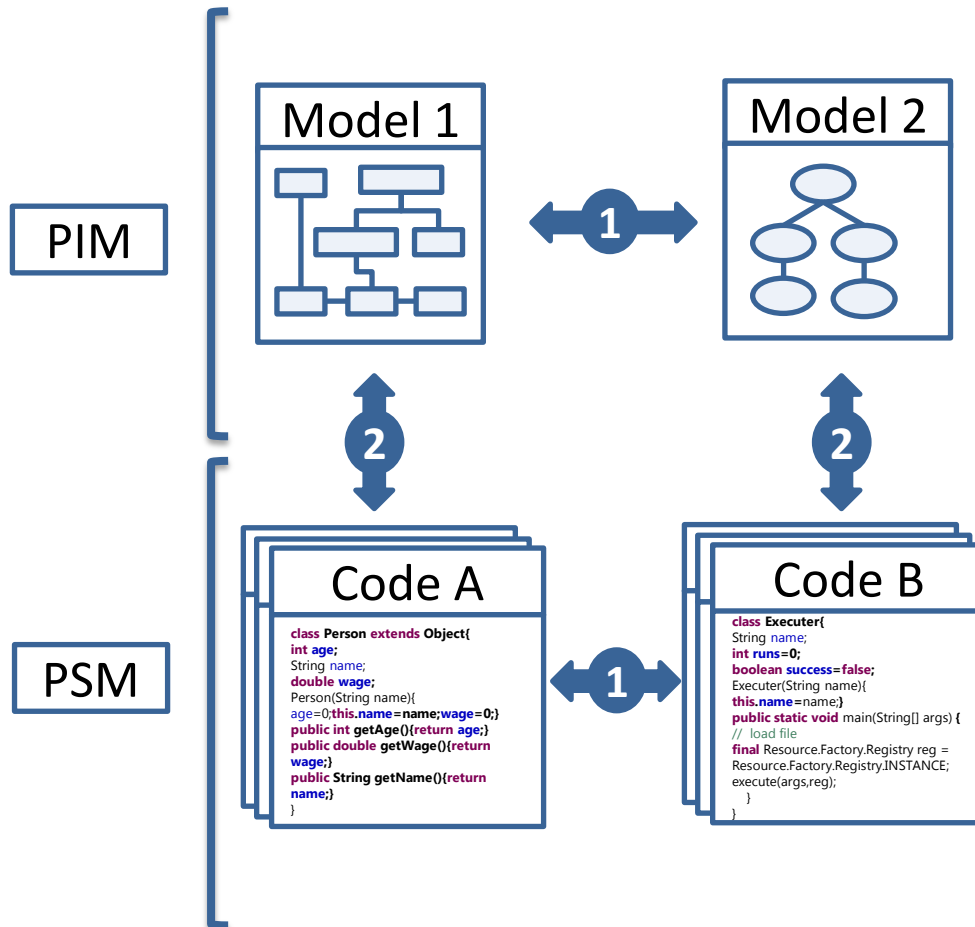


Model Transformations

- 1 PIM ↔ PIM (horizontal):**
DSL1 ↔ DSL2,
Model Translation/ Integration/
Synchronisation, e.g.:
UML Class Diag. ↔ RDBM
BPMN ↔ BPEL
Sequence Diag. ↔ St. Machines
- 2 PIM ↔ PSM (vertical):**
Model/Code generation,
reverse engineering, e.g.:
Class Diag. ↔ Class Diag.
Class Diag. ↔ Java

[HEO+13]

Interrelated Models in Model Driven Engineering



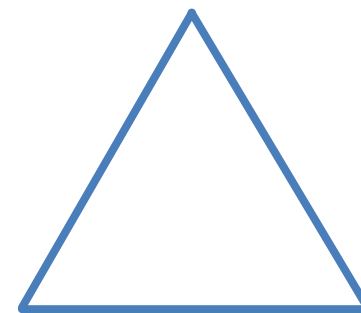
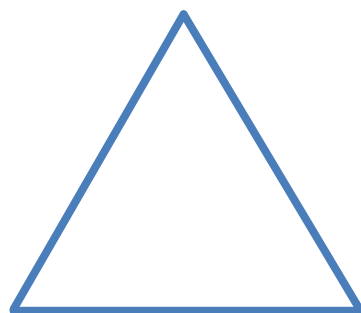
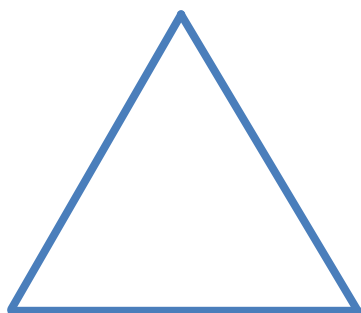
Model Transformations

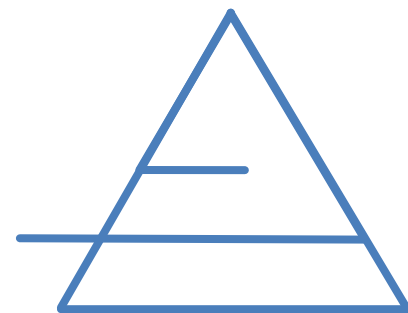
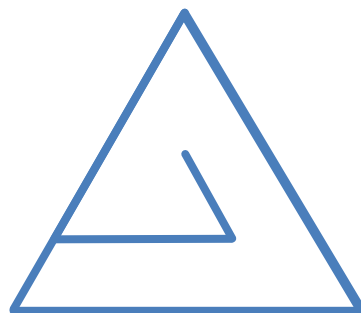
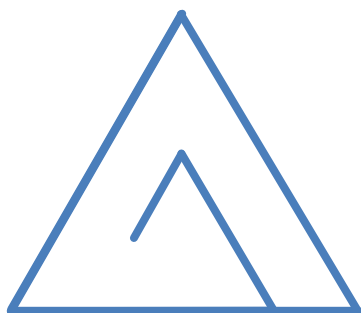
- 1 PIM ↔ PIM (horizontal):**
DSL1 ↔ DSL2,
Model Translation/ Integration/
Synchronisation, e.g.:
UML Class Diag. ↔ RDBM
BPMN ↔ BPEL
Sequence Diag. ↔ St. Machines
- 2 PIM ↔ PSM (vertical):**
Model/Code generation,
reverse engineering, e.g.:
Class Diag. ↔ Class Diag.
Class Diag. ↔ Java

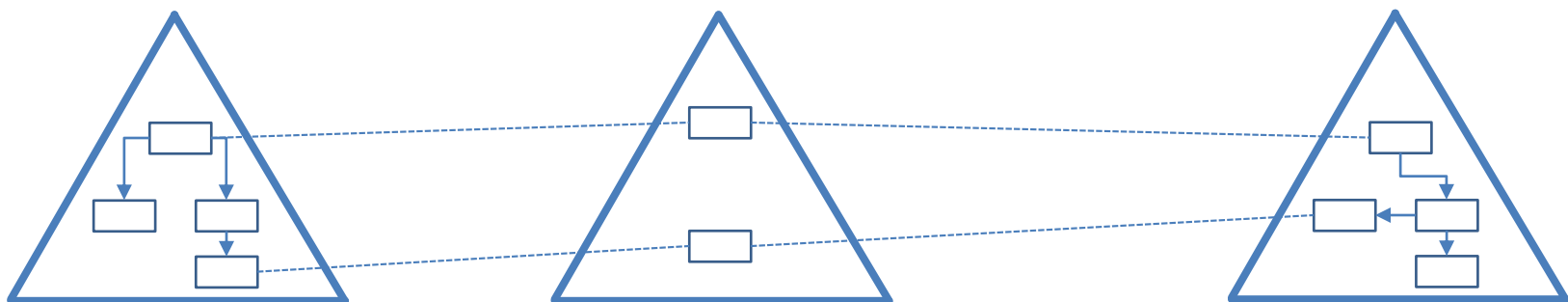
[HEO+13]





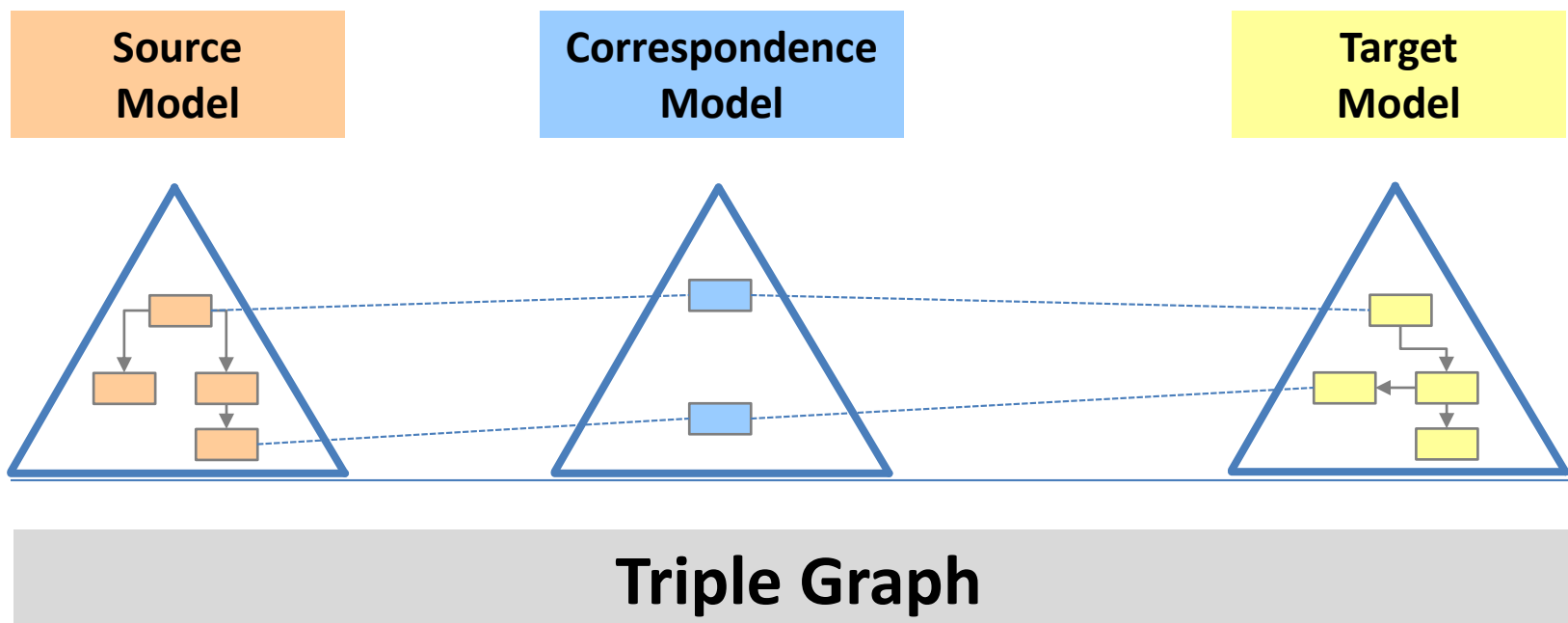






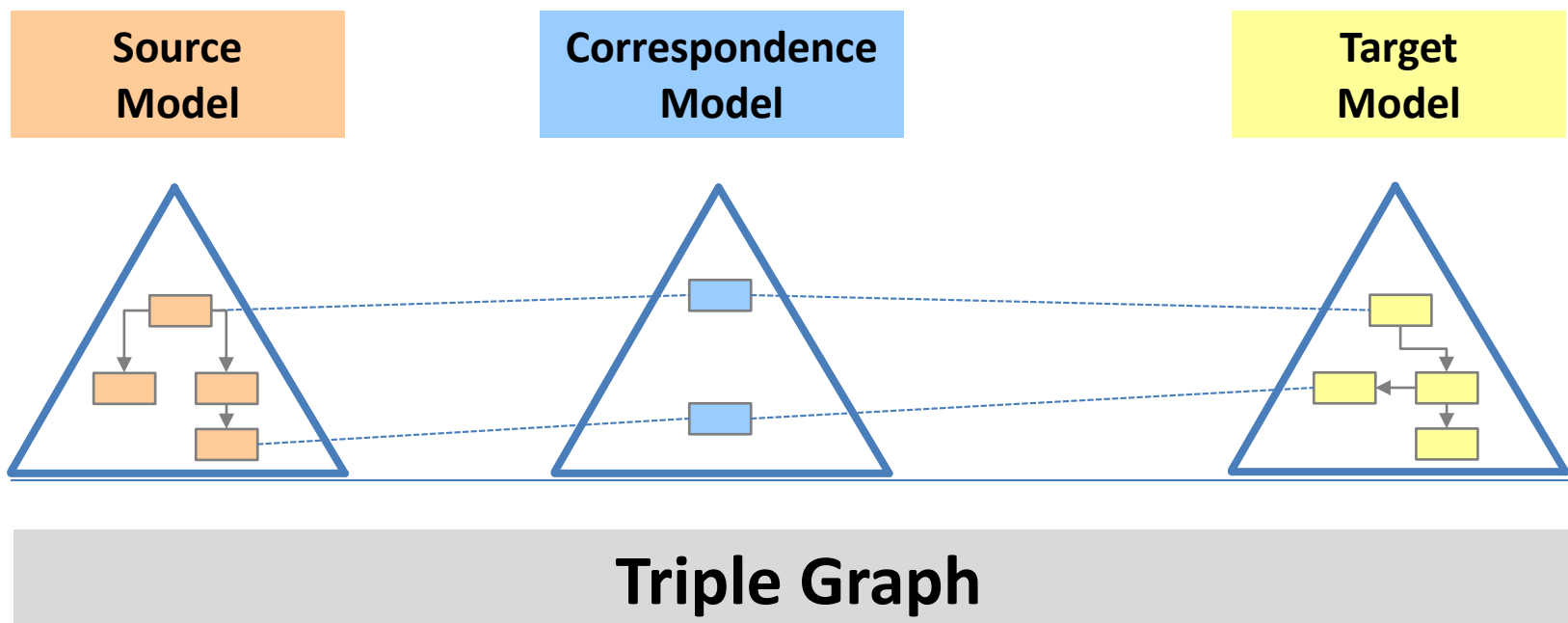
Key Idea of Triple Graph Grammars (TGGs)

- Specify pattern by pattern how **consistent integrated models** can be constructed **simultaneously**



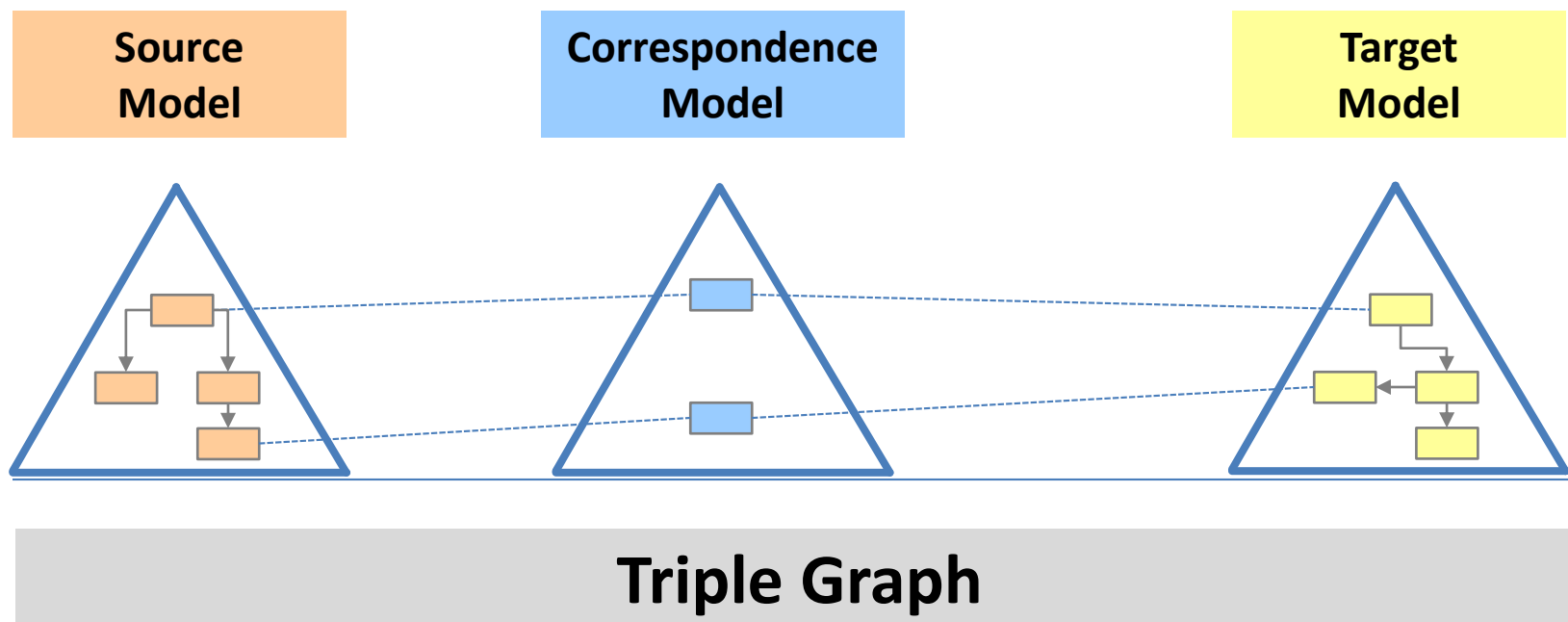
Key Idea of Triple Graph Grammars (TGGs)

- **Specify** pattern by pattern how **consistent integrated models** can be constructed **simultaneously**
- **Generate** operations for interoperability:
Model **Translation/Integration/Synchronisation**



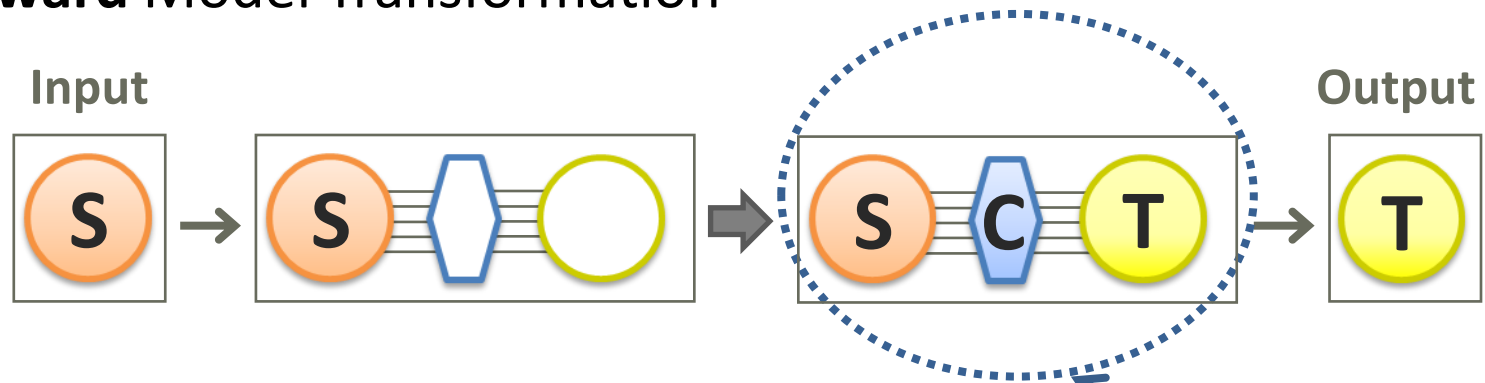
Key Idea of Triple Graph Grammars (TGGs)

- **Specify** pattern by pattern how **consistent integrated models** can be constructed **simultaneously**
- **Generate** operations for interoperability:
Model **Translation/Integration/Synchronisation**

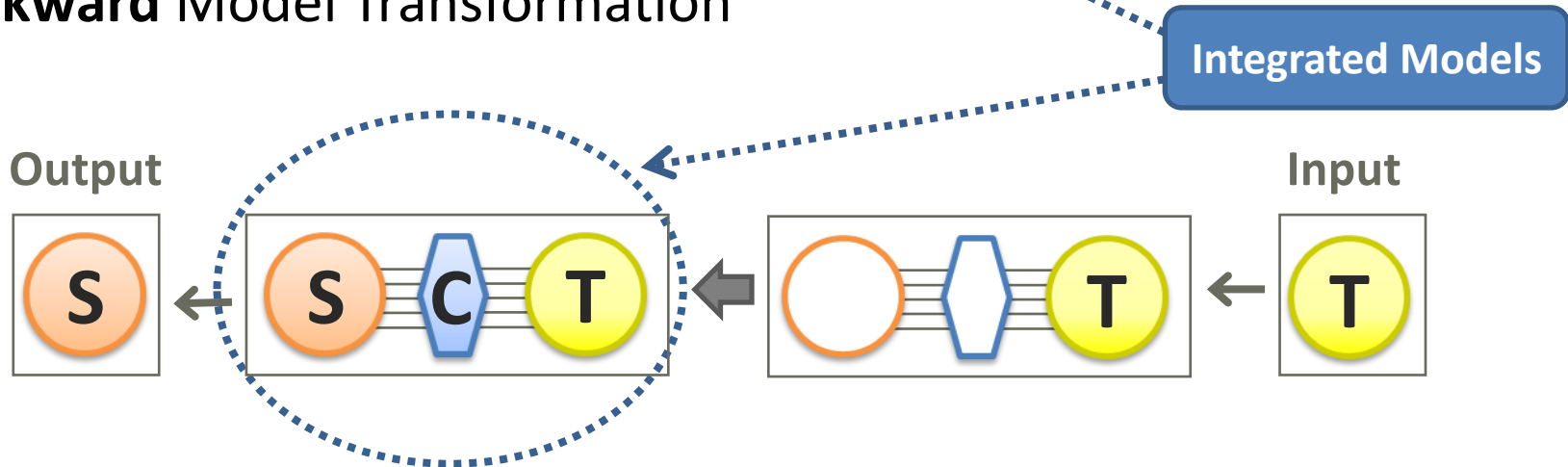


Bidirectional Model Transformations Based on TGGs: Concept

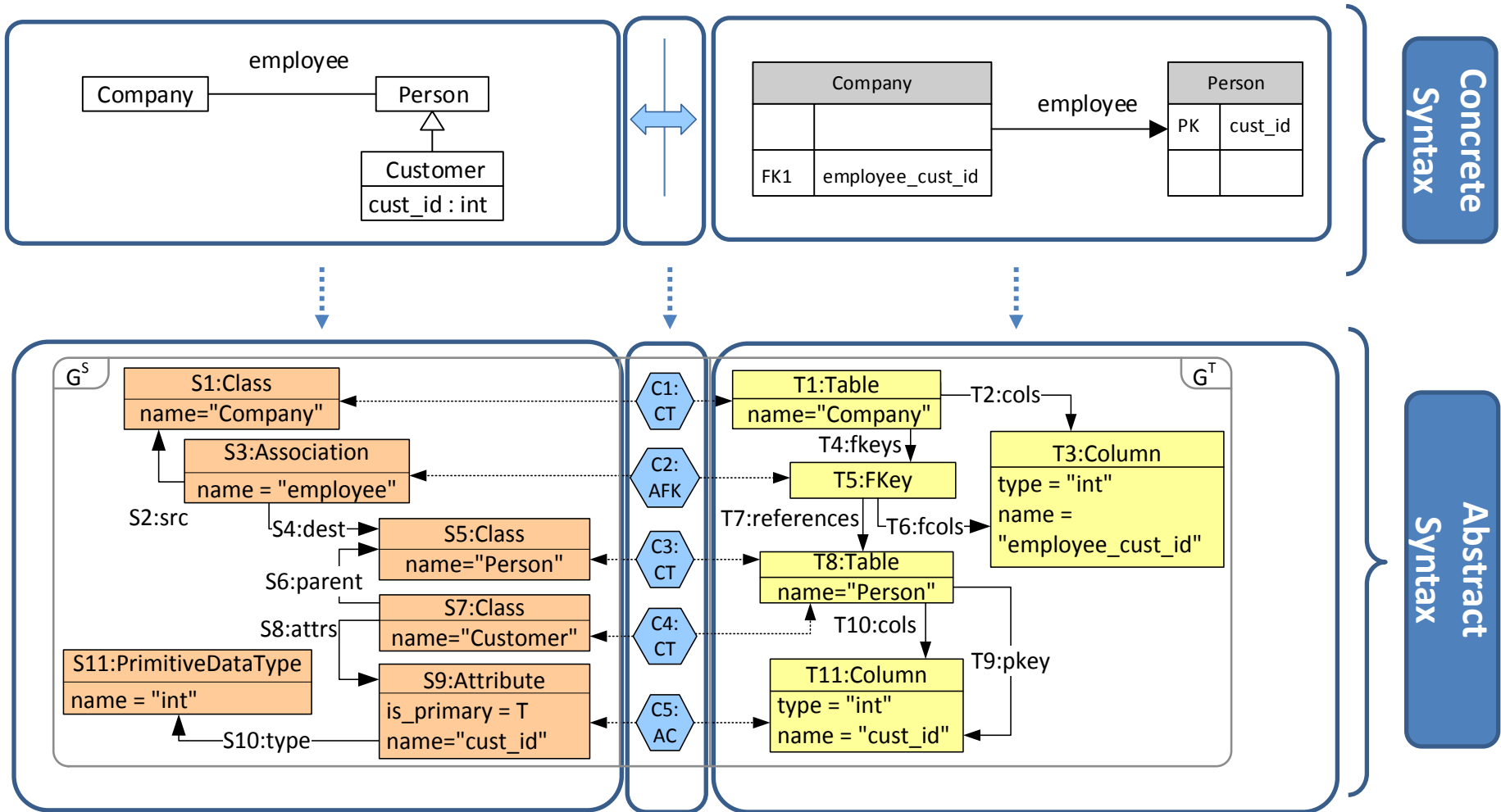
Forward Model Transformation



Backward Model Transformation



CD2RDBM: Integrated Model

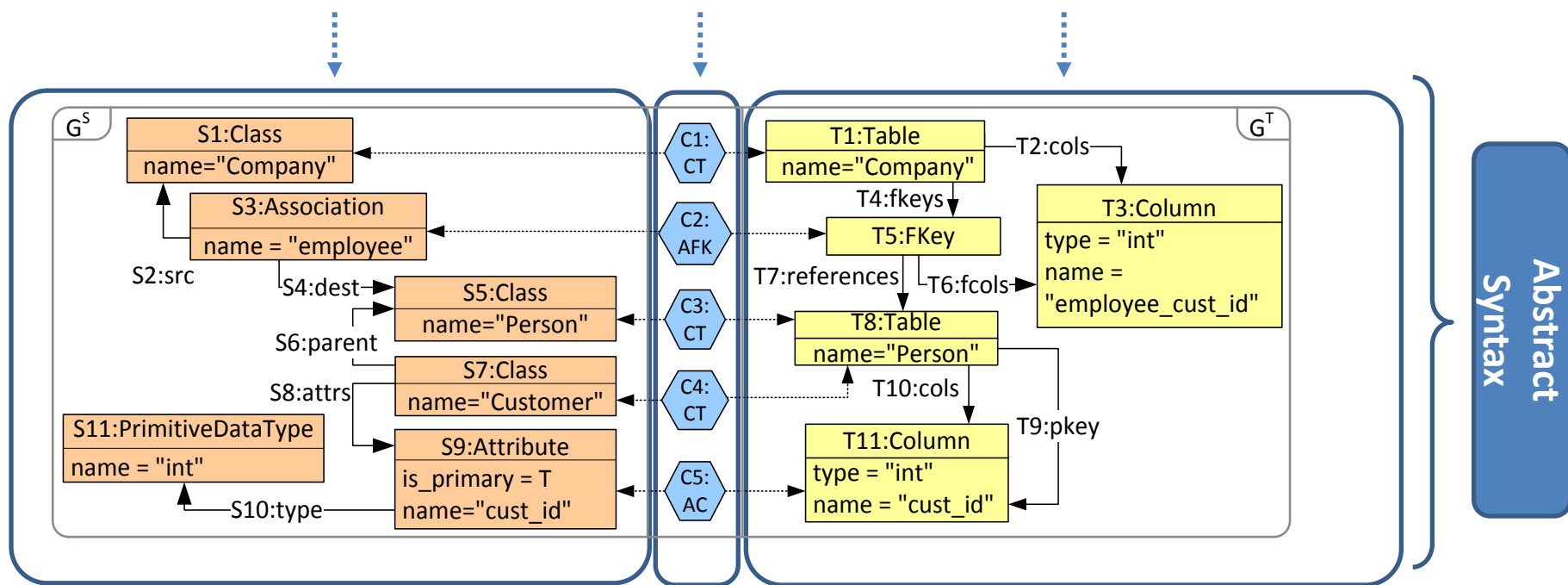


CD2RDBM: Integrated Model

Source Model

Correspondence Model

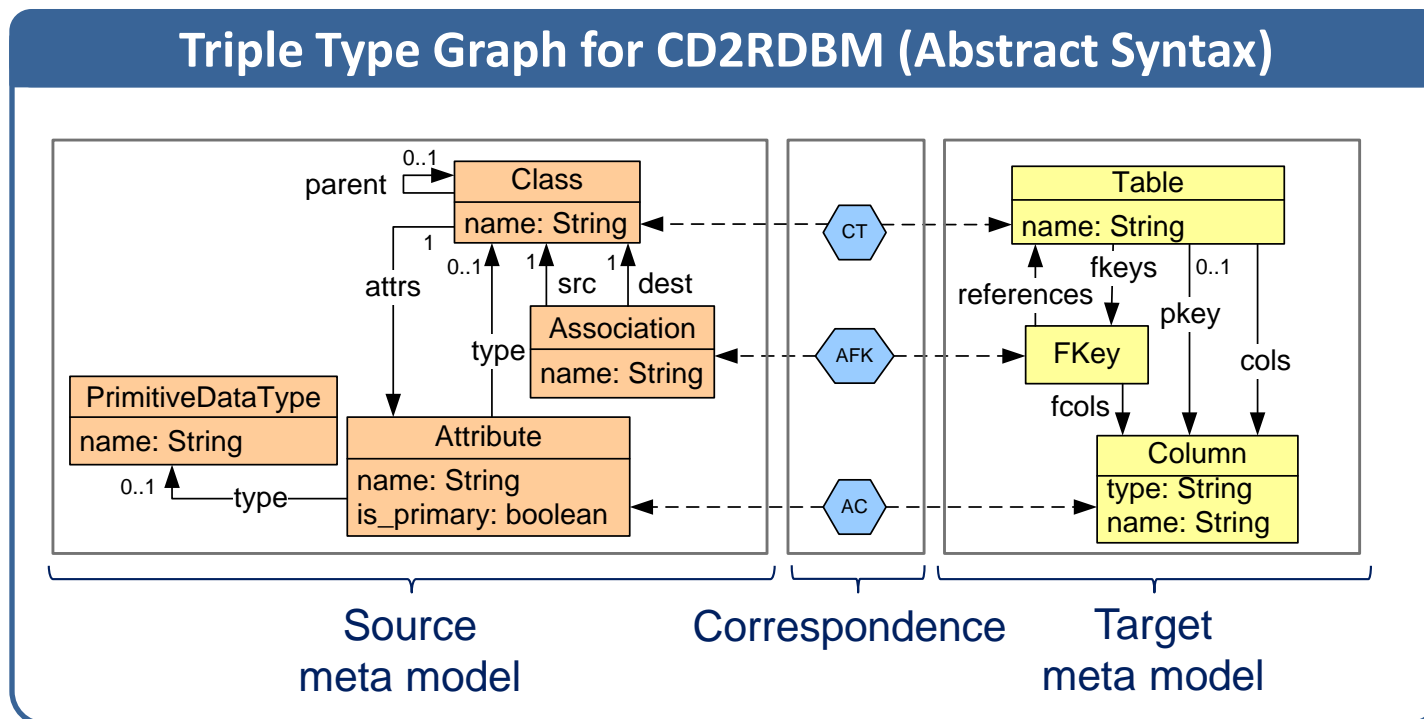
Target Model



Triple Graph

Triple Graph Grammar (TGG)

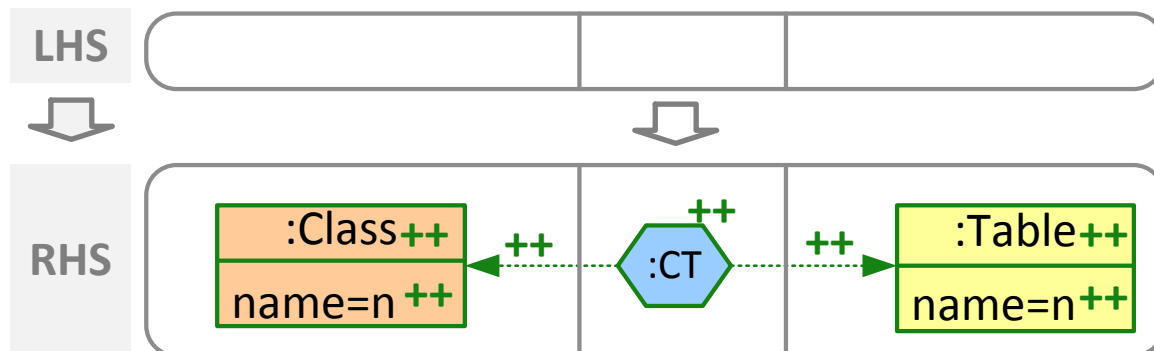
- $TGG = (TG, SG, TR)$,
 TG = type graph, SG = start graph ($SG = \emptyset$), TR = set of triple rules



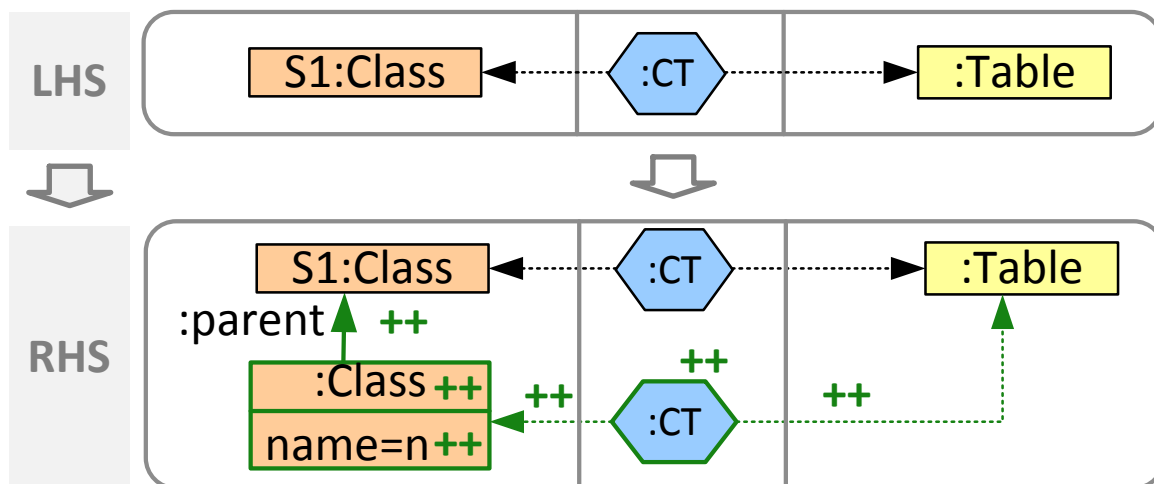
[Schürr94]

2 Triple Rules for CD2RDBM

Class2Table(n:String)



Subclass2Table(n:String)

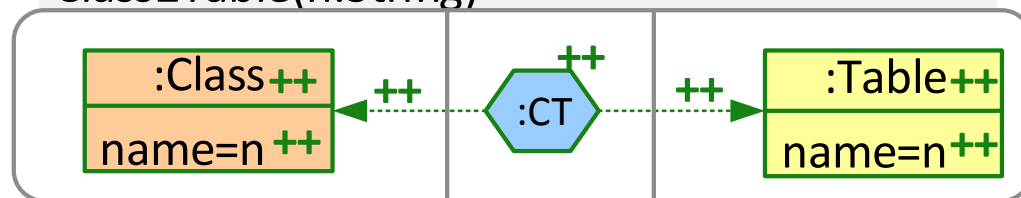


2 Triple Rules for CD2RDBM (Compact)

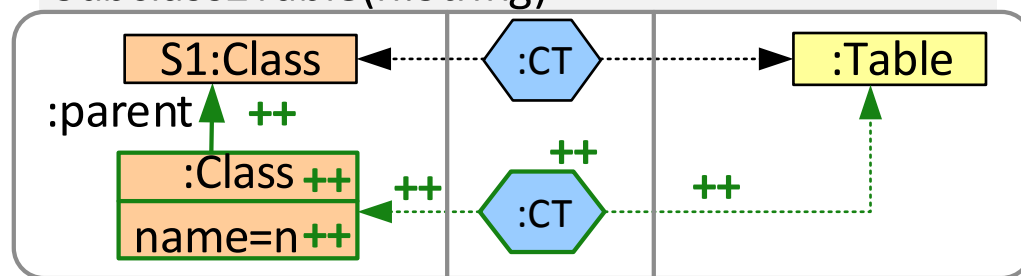
Compact Notation:

- L and R withing one figure
- elements in $R \setminus L$ are marked with "++"

Class2Table($n:String$)



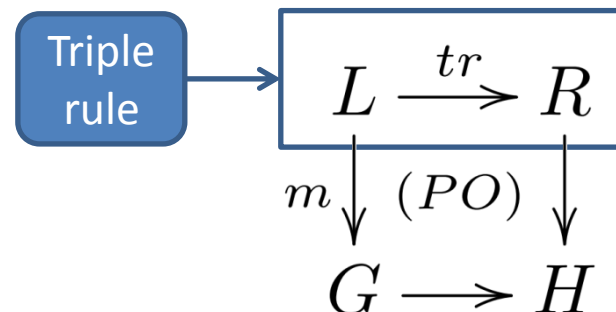
Subclass2Table($n:String$)



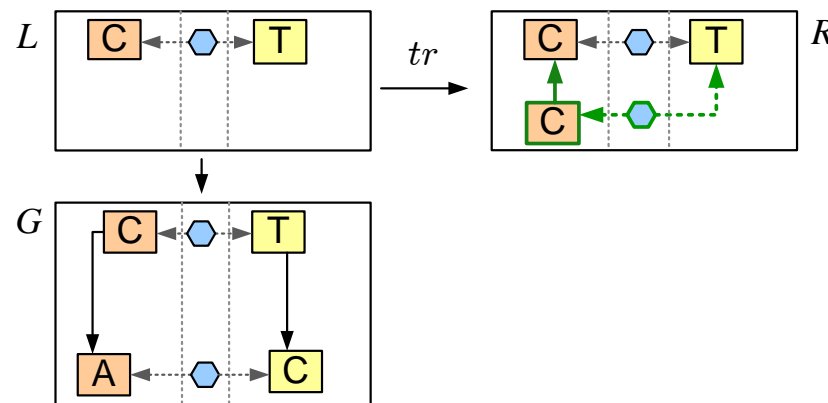
Transformation Step: General Concept

FORMAL:

construction is a pushout (PO) of tr and m in the category of triple graphs



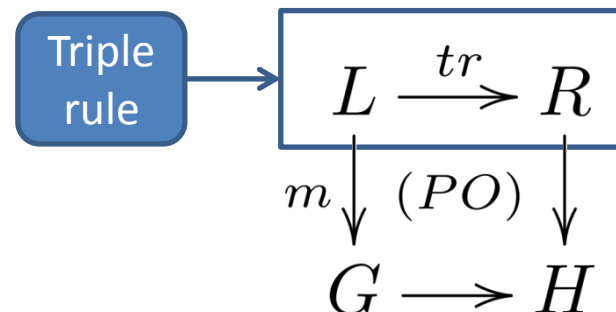
1. Pattern matching:
Find occurrence $m(L)$ of L in G



Transformation Step: General Concept

FORMAL:

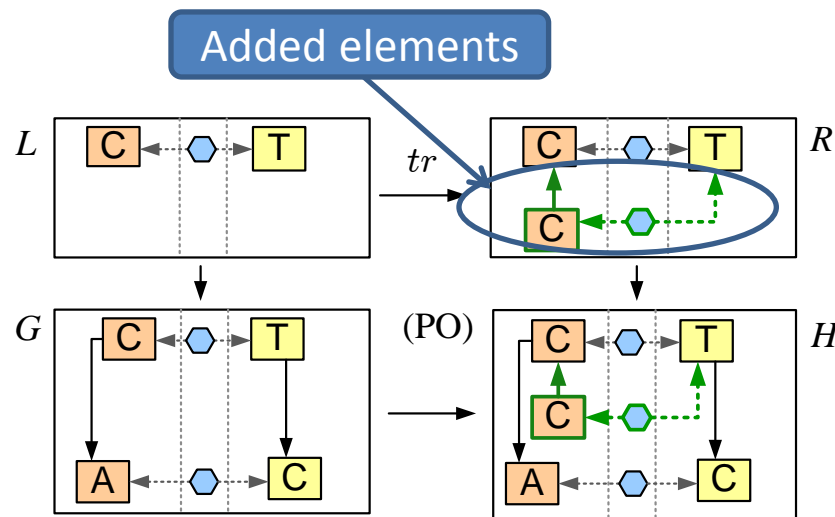
construction is a pushout (PO) of tr and m in the category of triple graphs



1. Pattern matching:
Find occurrence $m(L)$ of L in G
2. Transformation:
Replace $m(L)$ in G by R

RESULT:

Transformation Step $G \xRightarrow{tr, m} H$

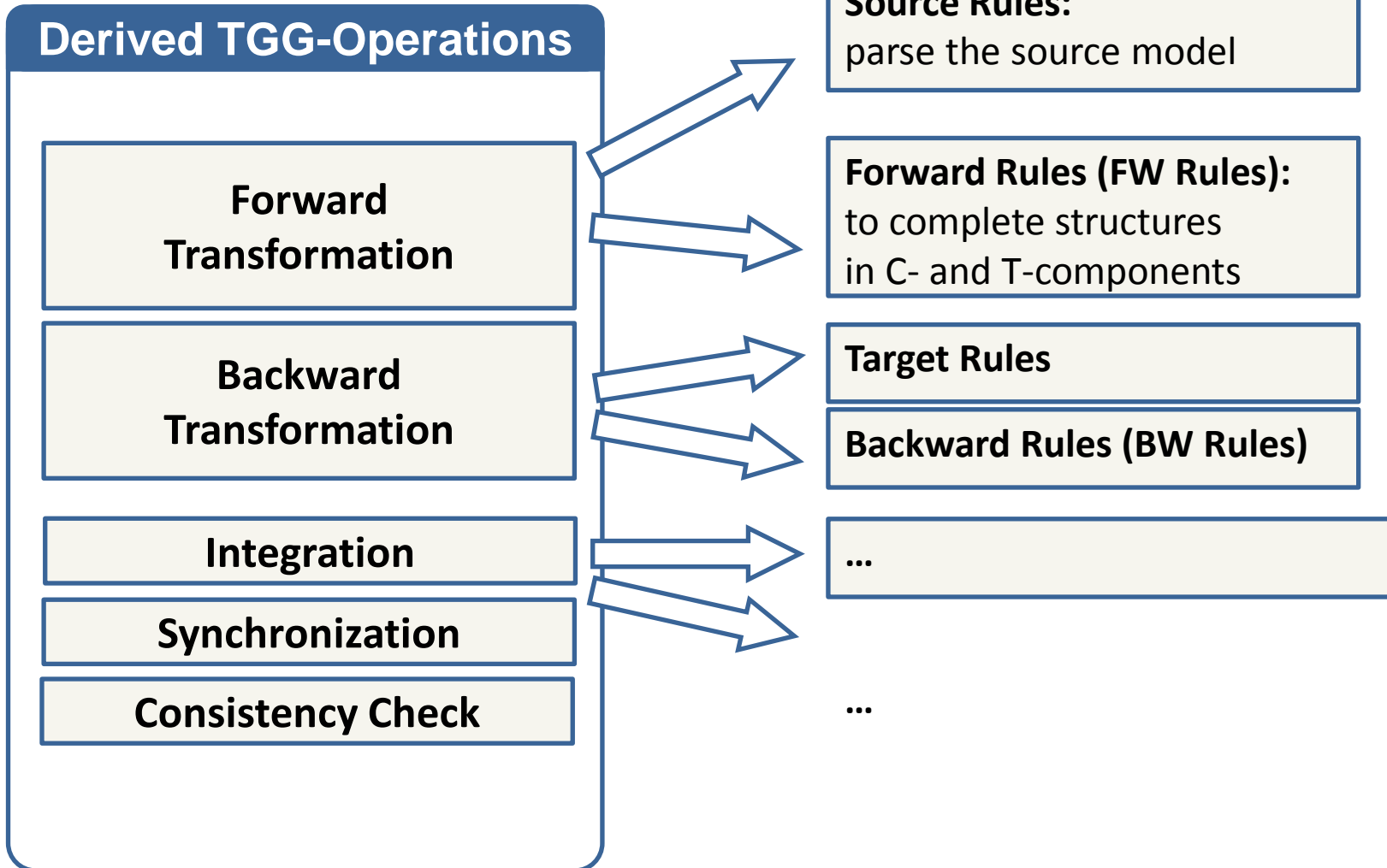


Language of Integrated Models

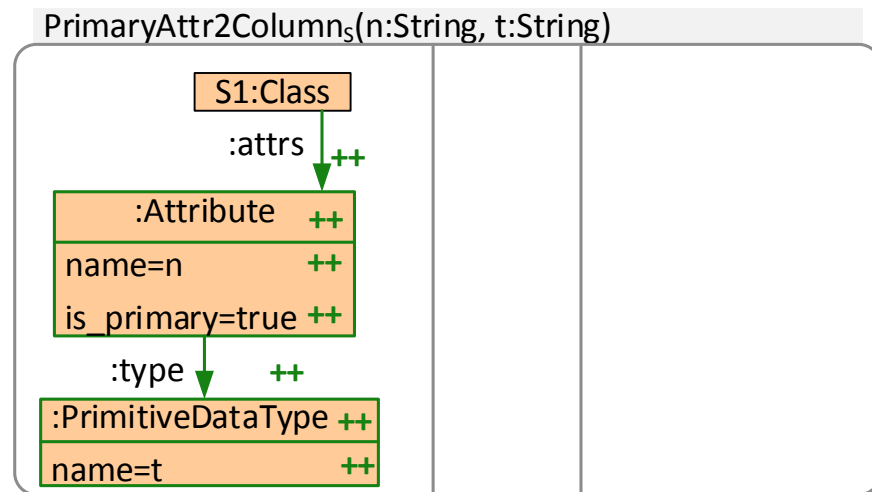
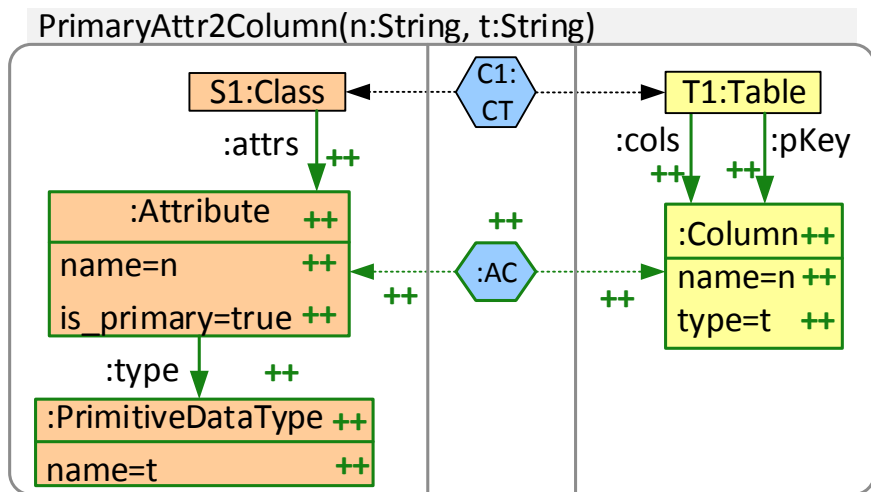
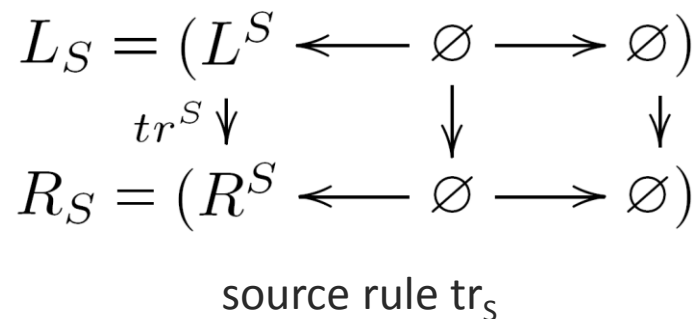
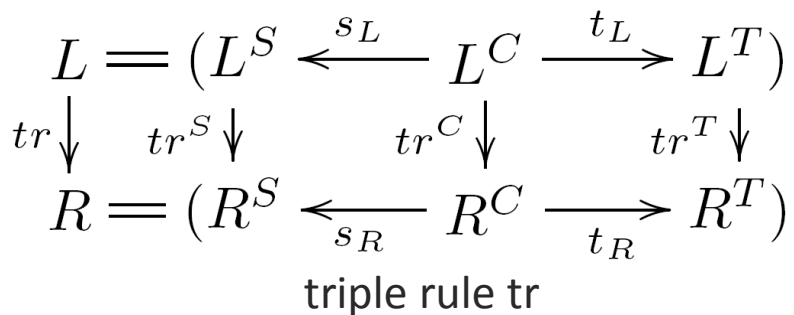
- A **triple rule** specifies how model elements in the source and target language can be created **synchronously**
- Each **triple rule** is interpreted as **consistency pattern**: a source and a target model are consistent, if they can be constructed by the TGG
- A **TGG** generates the **language of consistent integrated models**:

$$L(TGG) = \{G \mid \emptyset \Rightarrow^* G \text{ via } TR\}$$

Operational Rules

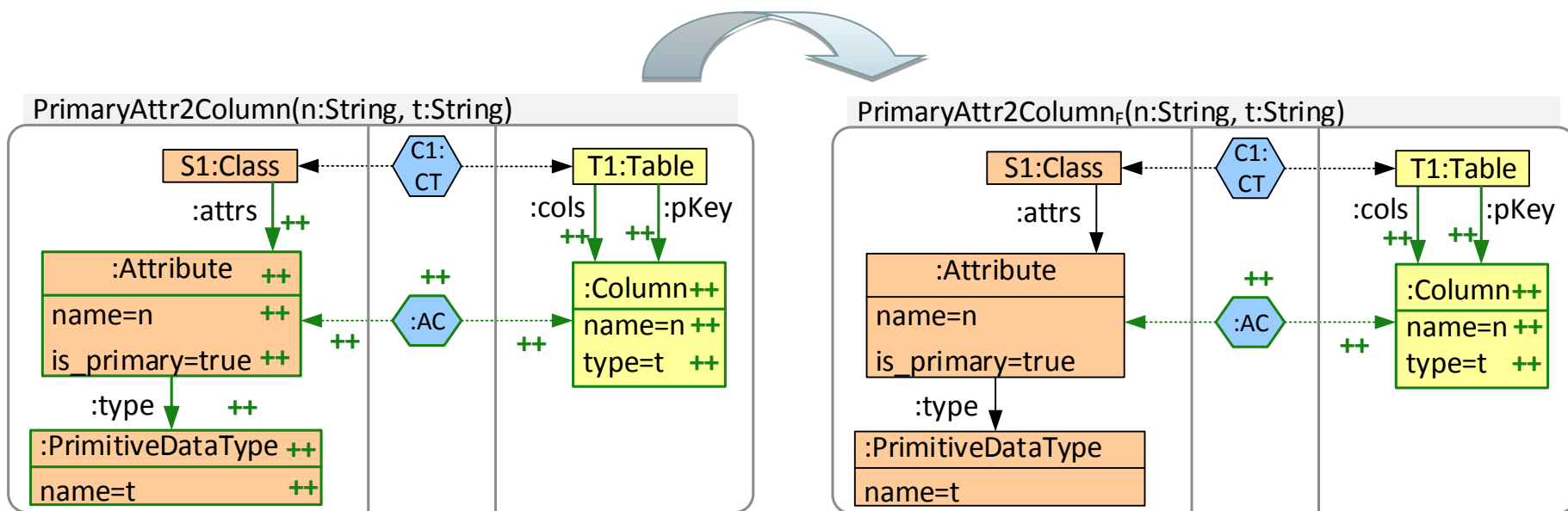


Derived Source Rule

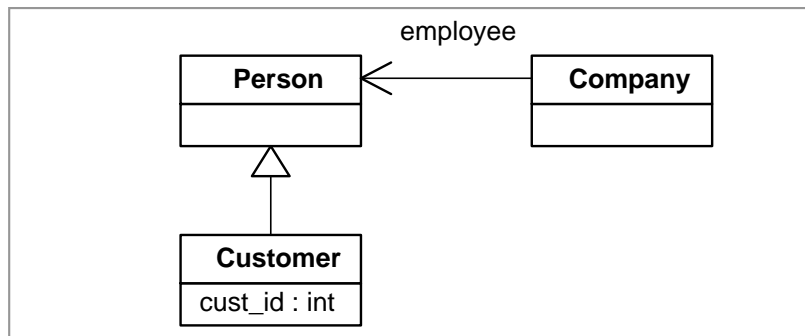


Derived Forward Rule

$$\begin{array}{ccc}
 L = (L^S \xleftarrow{s_L} L^C \xrightarrow{t_L} L^T) & & L_F = (R^S \xleftarrow{tr^S \circ s_L} L^C \xrightarrow{t_L} L^T) \\
 \begin{array}{cccc}
 tr \downarrow & tr^S \downarrow & tr^C \downarrow & tr^T \downarrow \\
 R = (R^S \xleftarrow{s_R} R^C \xrightarrow{t_R} R^T) & \Rightarrow & R_F = (R^S \xleftarrow{s_R} R^C \xrightarrow{t_R} R^T) & \\
 \text{triple rule tr} & & \text{forward rule tr}_F &
 \end{array}
 \end{array}$$

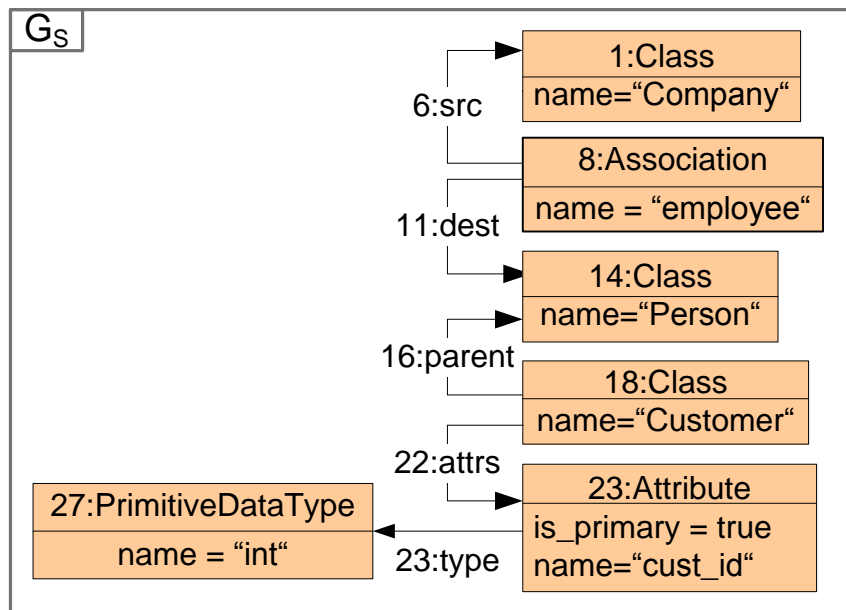


Example: Translation of Class Diagram

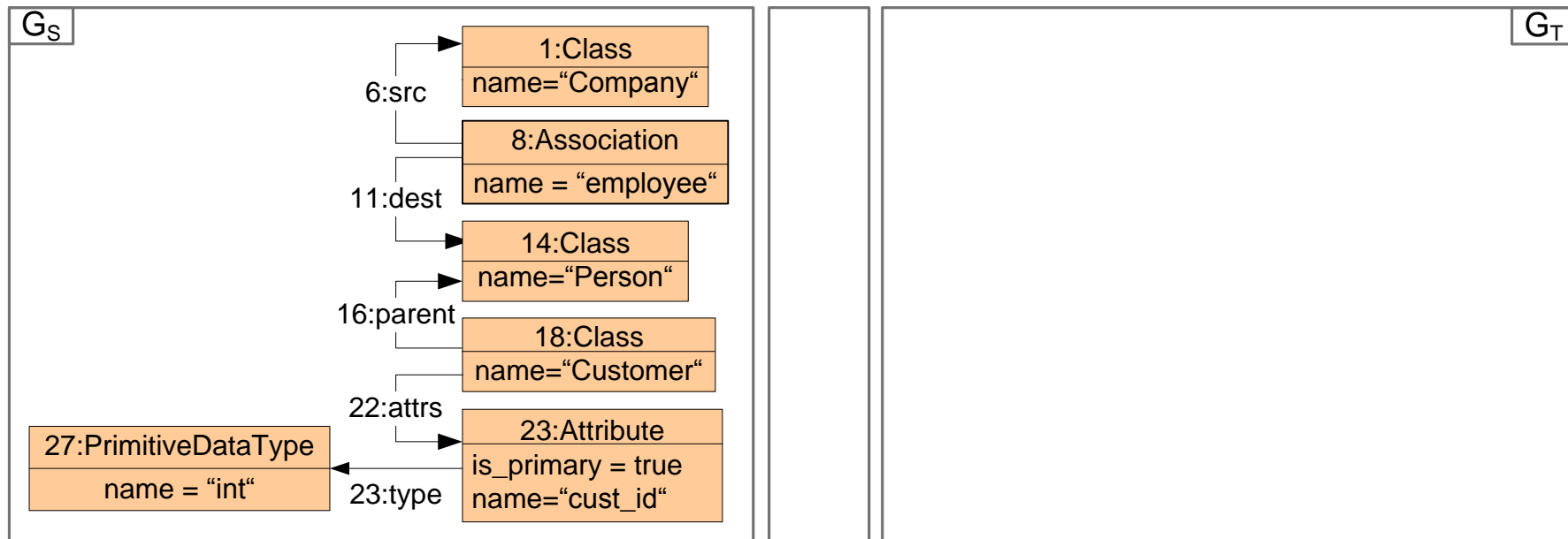
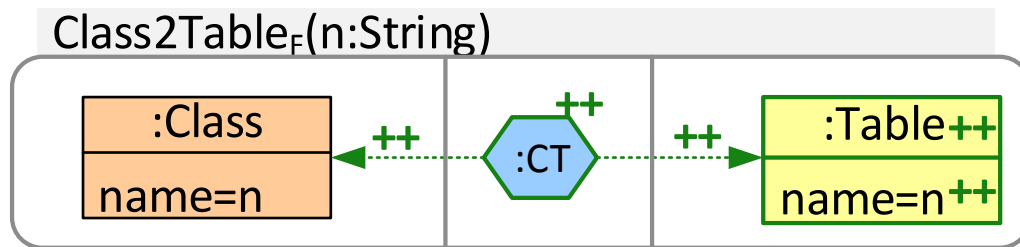
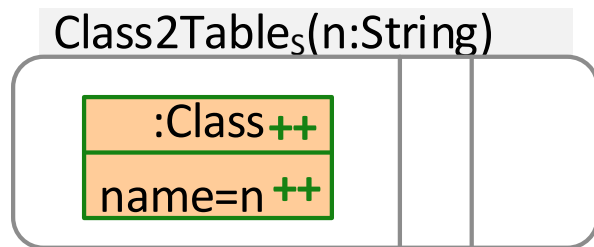


concrete syntax

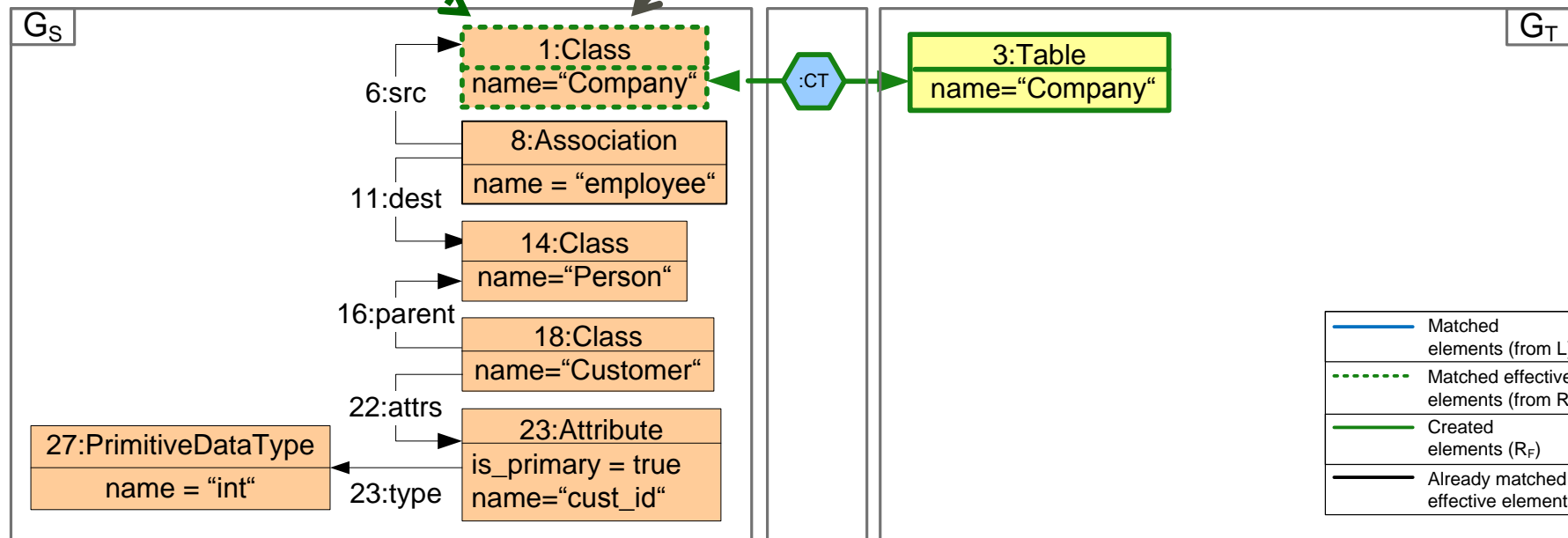
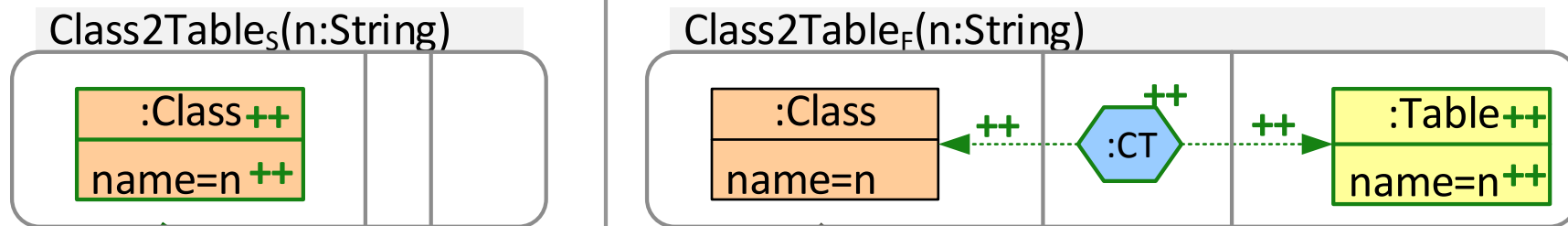
abstract syntax



Example: Translation of Class Diagram

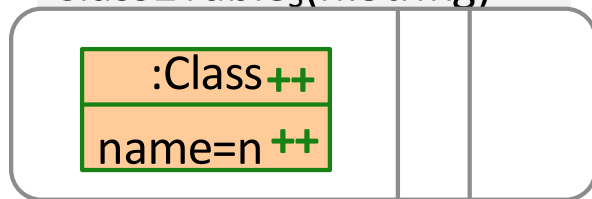


Example: Translation of Class Diagram

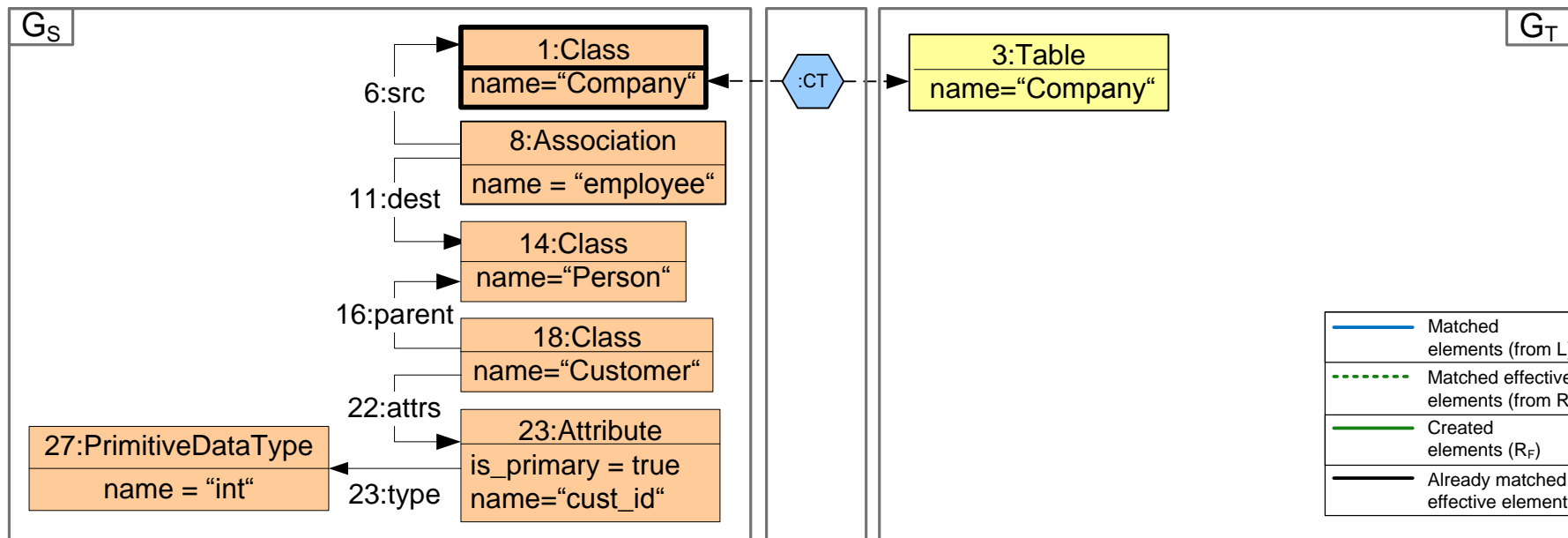
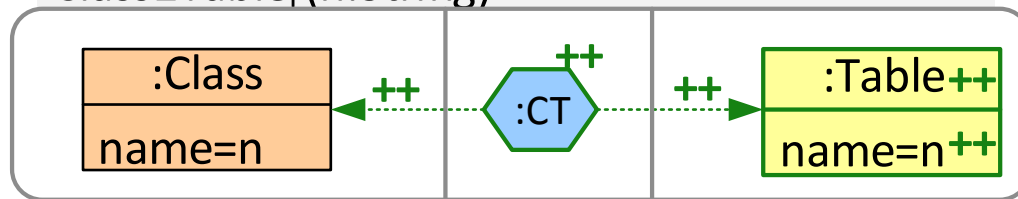


Example: Translation of Class Diagram

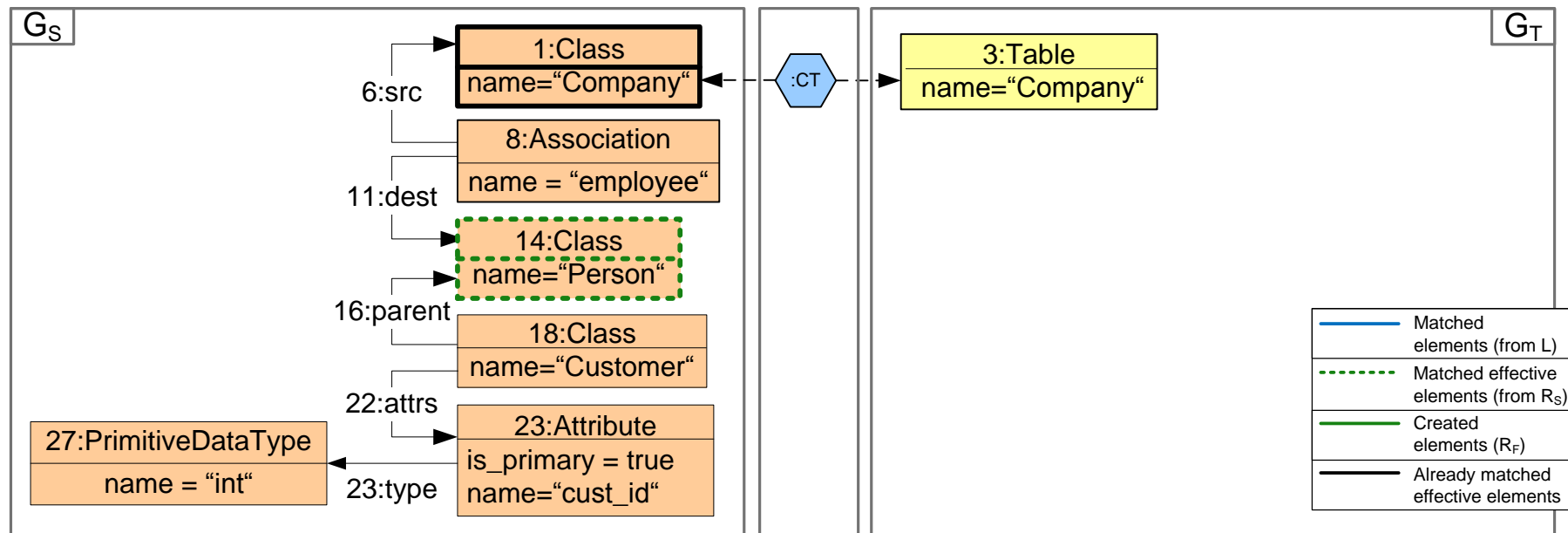
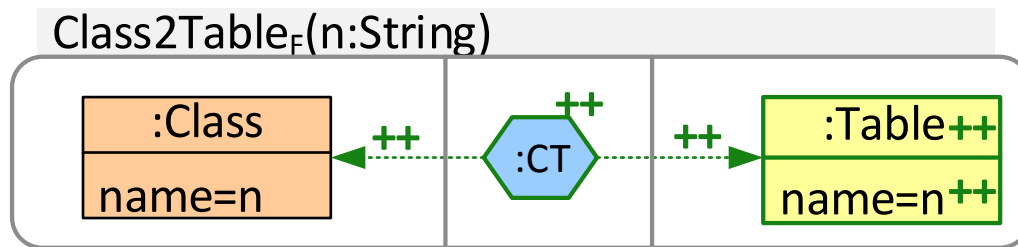
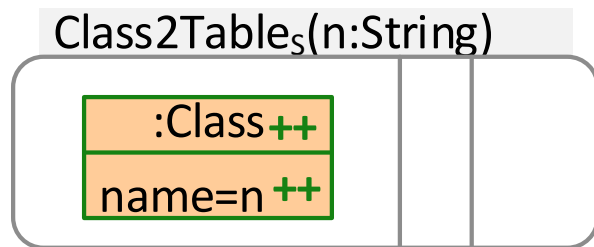
Class2Table_S(n:String)



Class2Table_F(n:String)

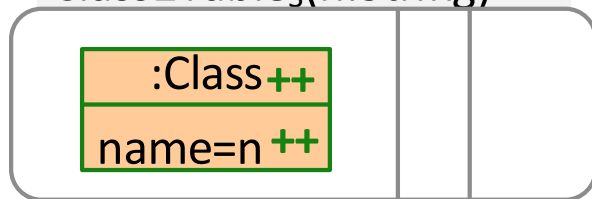


Example: Translation of Class Diagram

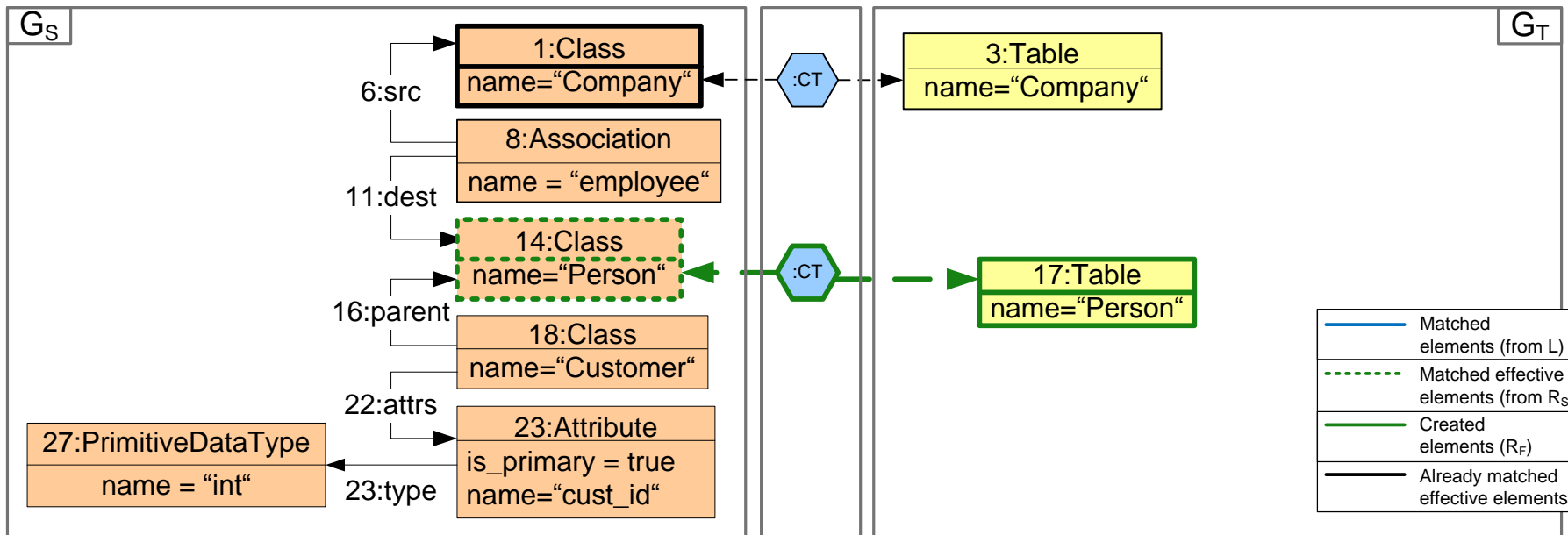
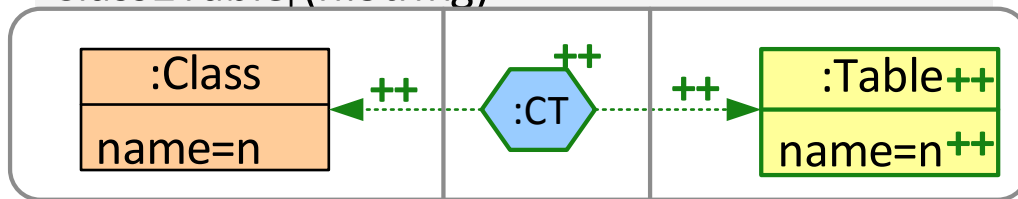


Example: Translation of Class Diagram

Class2Table_S(n:String)

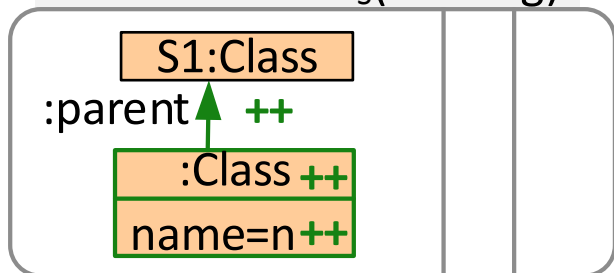


Class2Table_F(n:String)

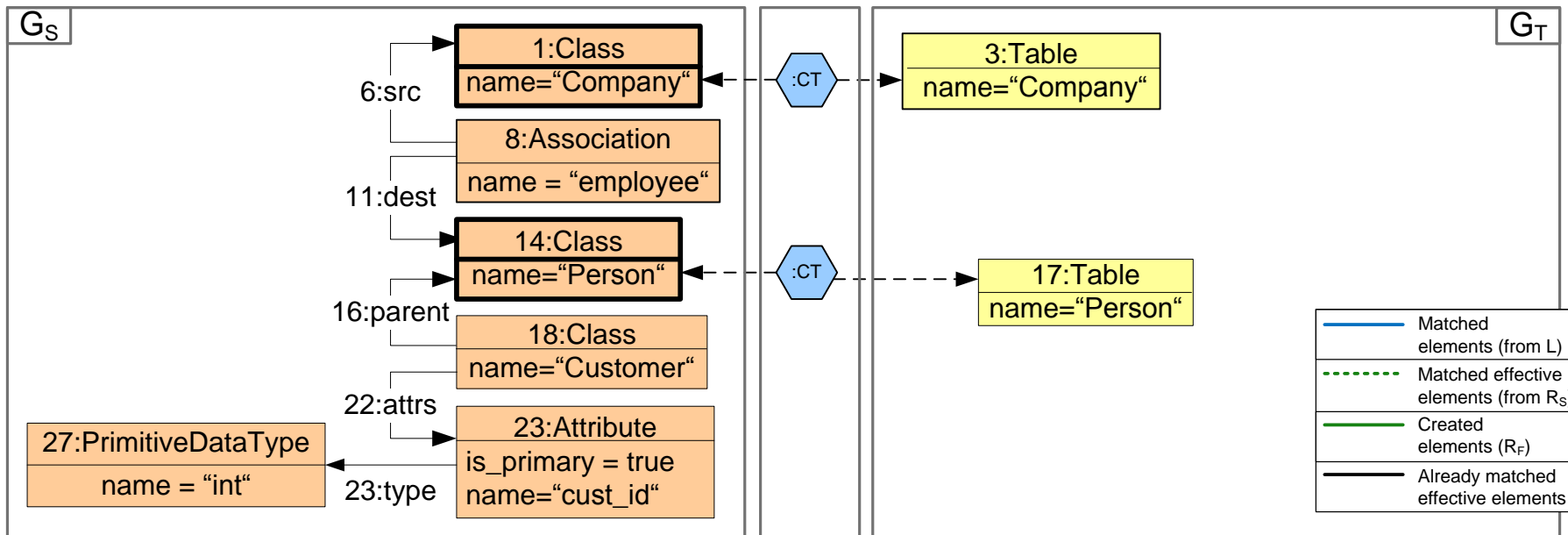
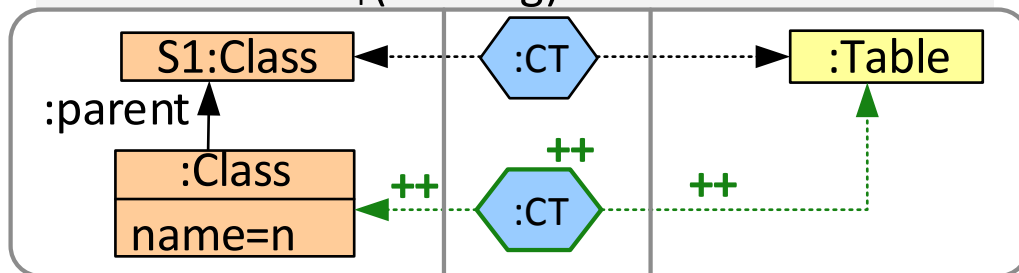


Example: Translation of Class Diagram

Subclass2Table_S(n:String)

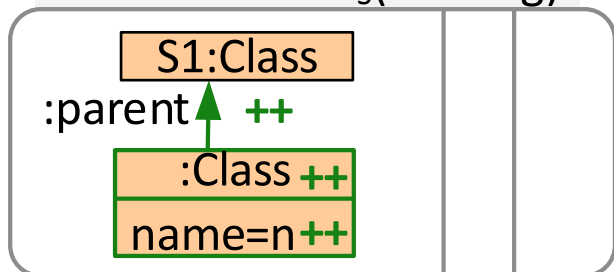


Subclass2Table_F(n:String)

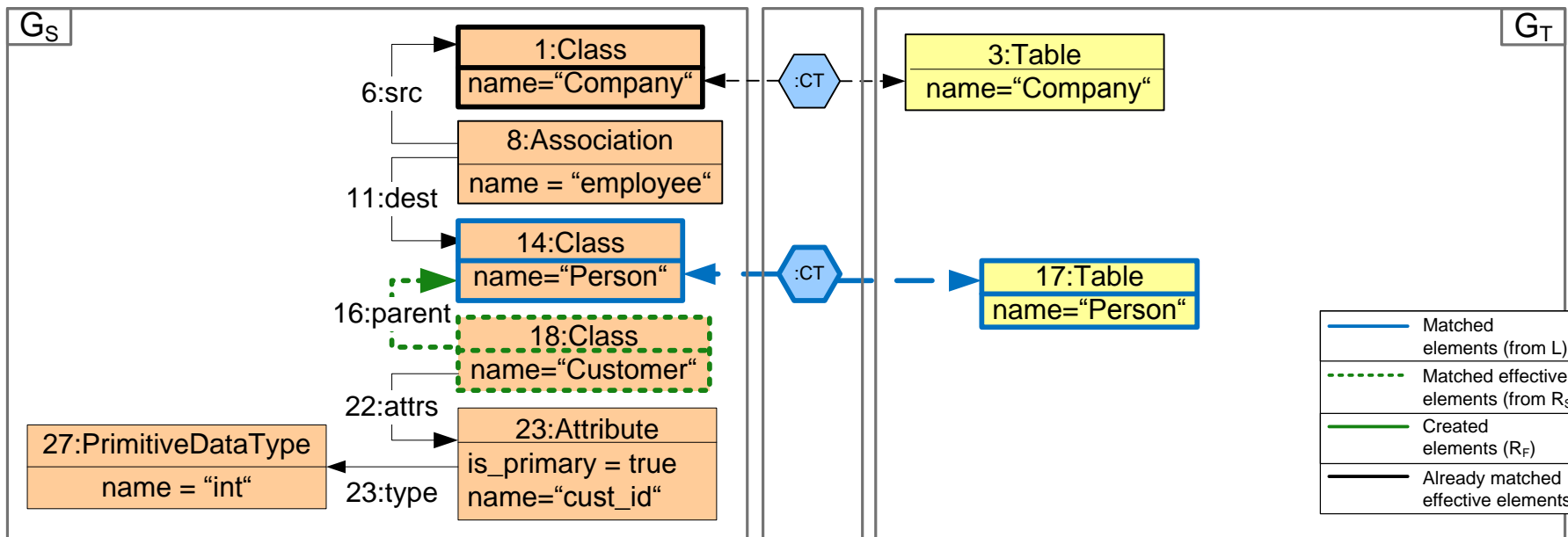
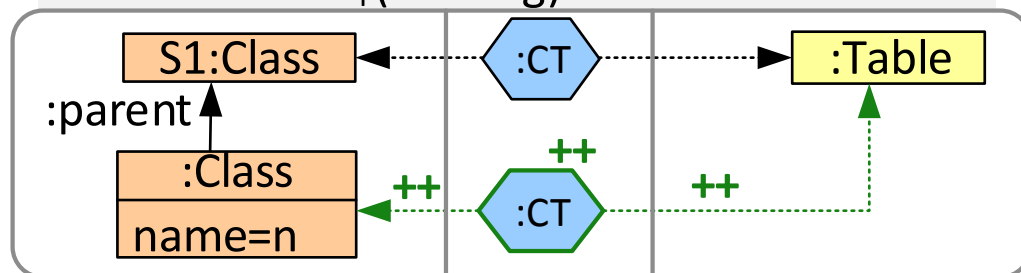


Example: Translation of Class Diagram

Subclass2Table_S(n:String)

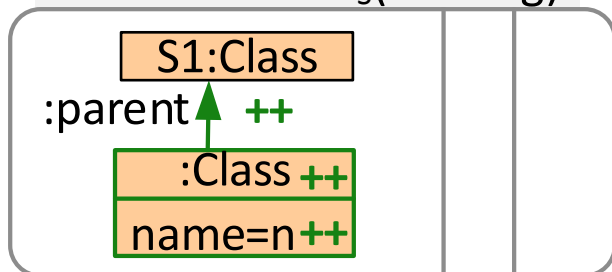


Subclass2Table_F(n:String)

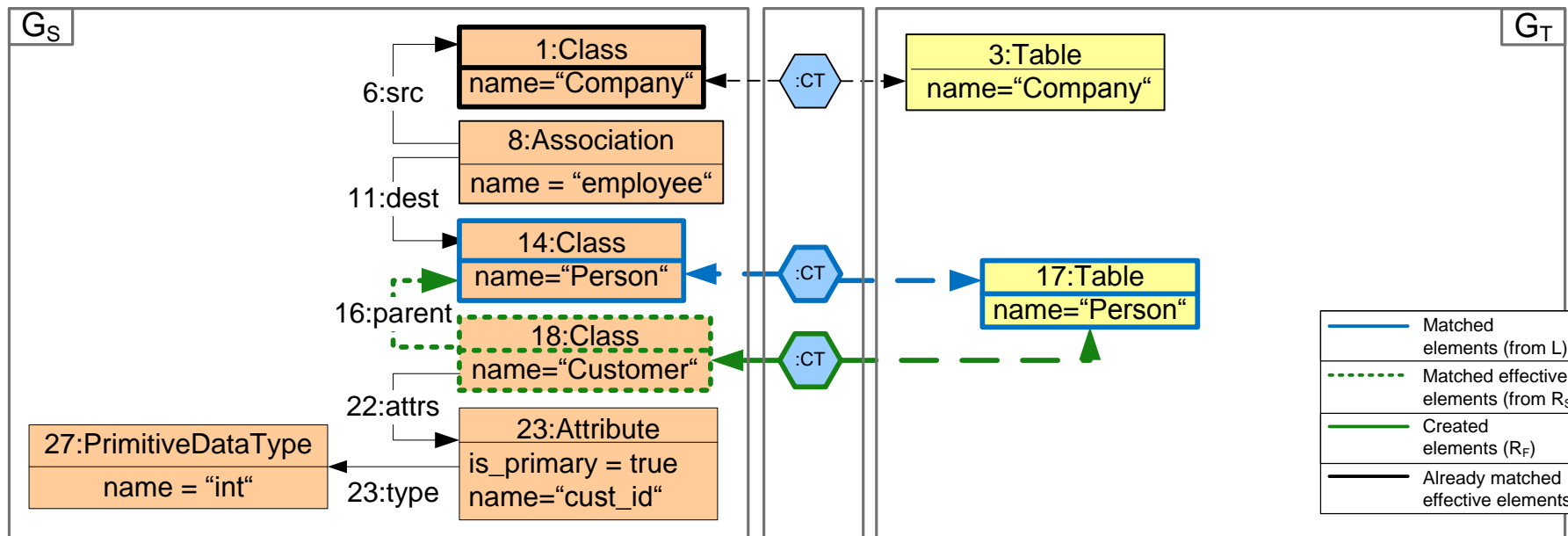
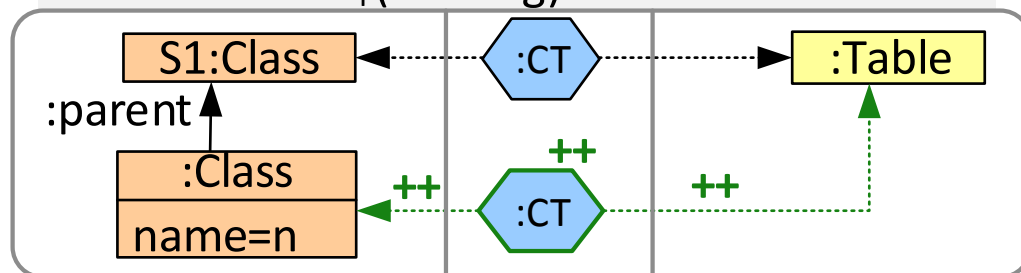


Example: Translation of Class Diagram

Subclass2Table_S(n:String)

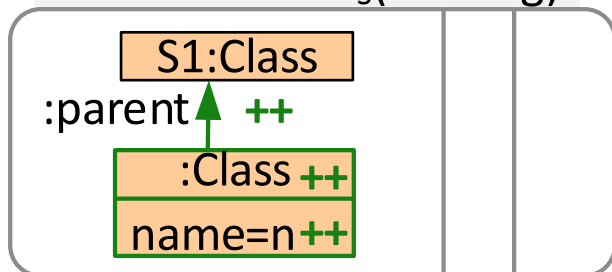


Subclass2Table_F(n:String)

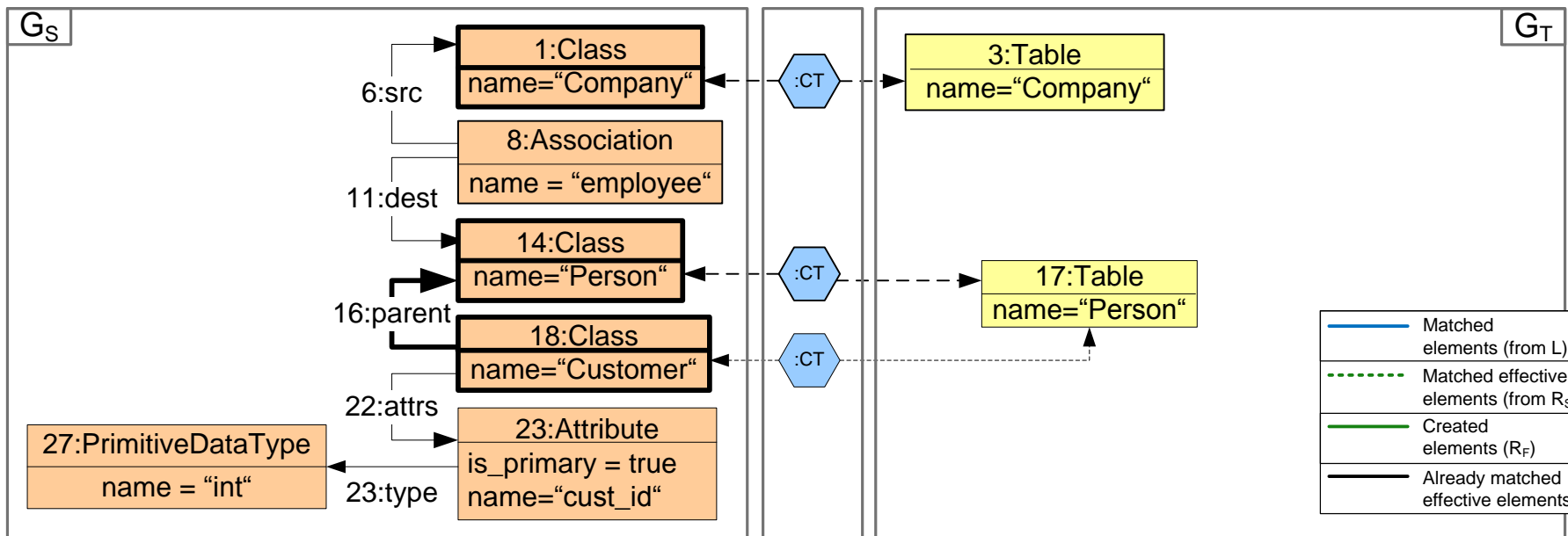
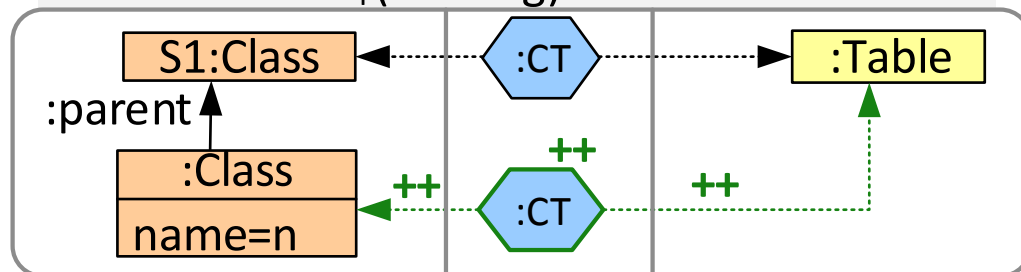


Example: Translation of Class Diagram

Subclass2Table_S(n:String)

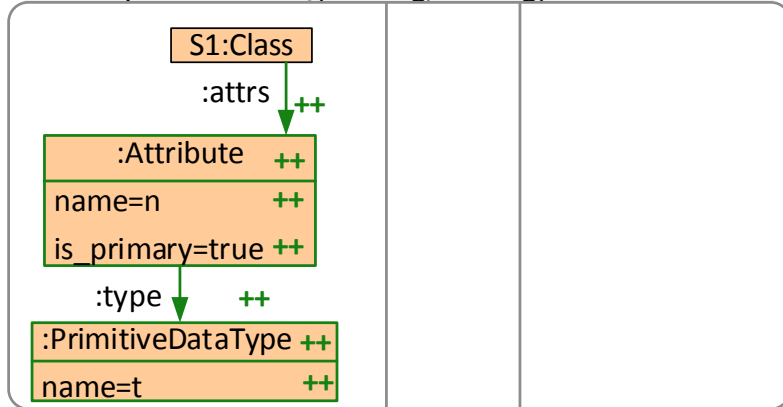


Subclass2Table_F(n:String)

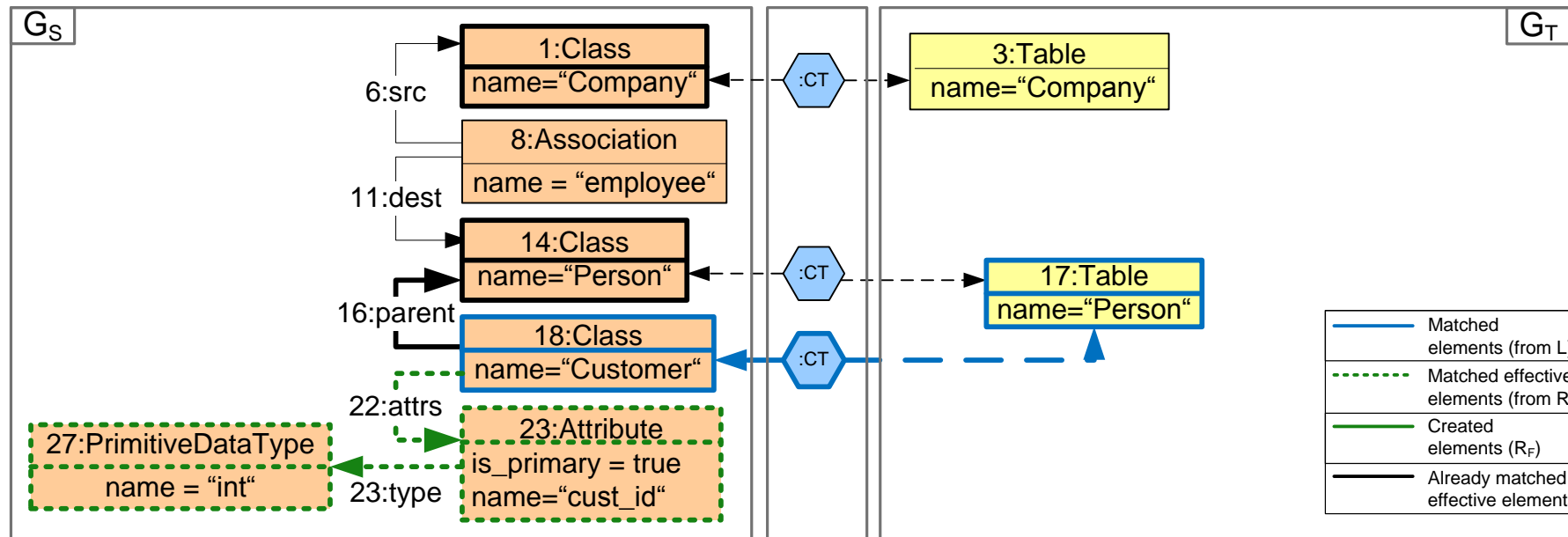
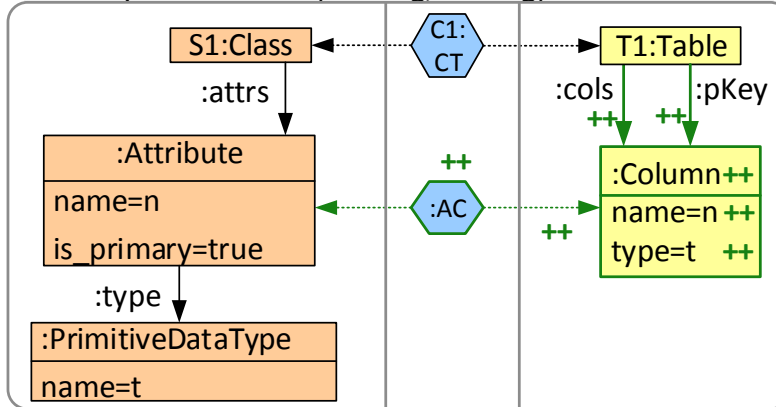


Example: Translation of Class Diagram

PrimaryAttr2Column_s(n:String, t:String)

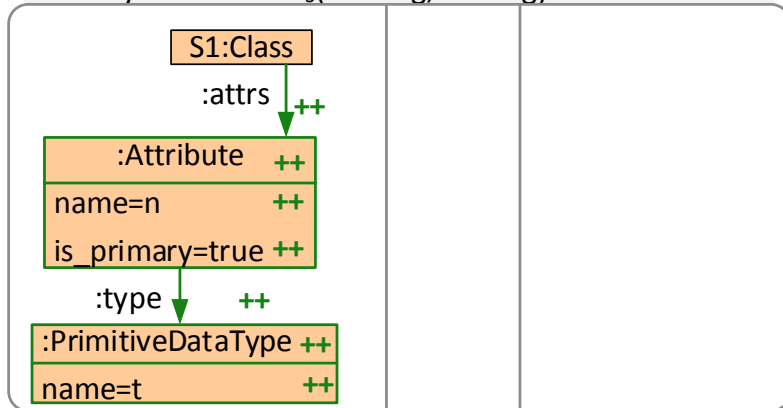


PrimaryAttr2Column_r(n:String, t:String)

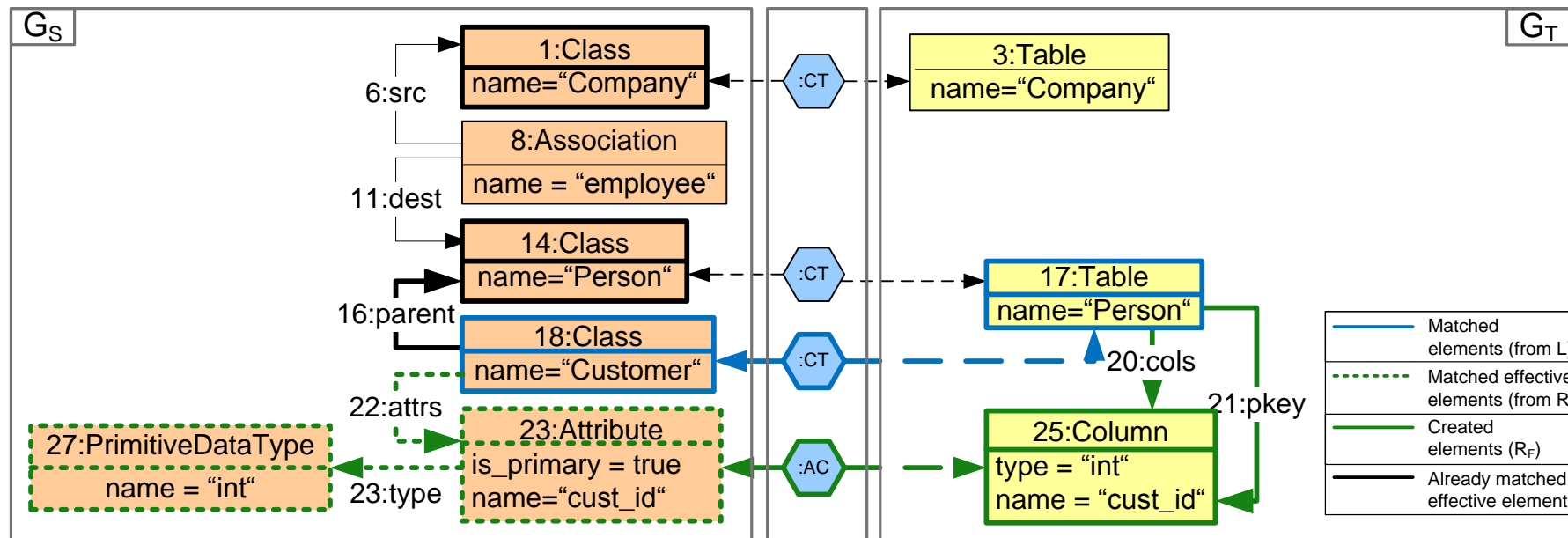
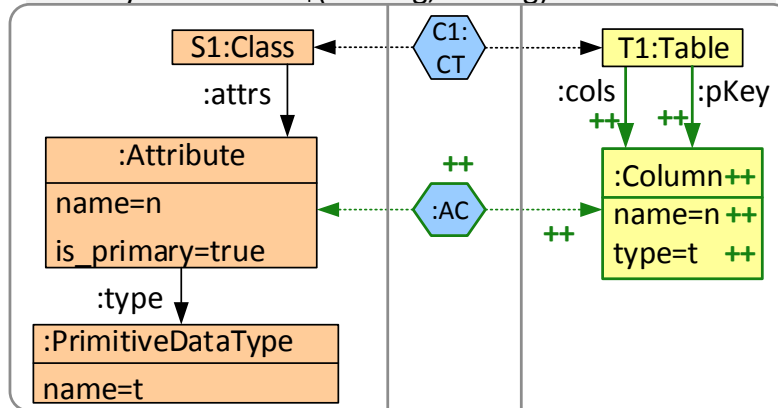


Example: Translation of Class Diagram

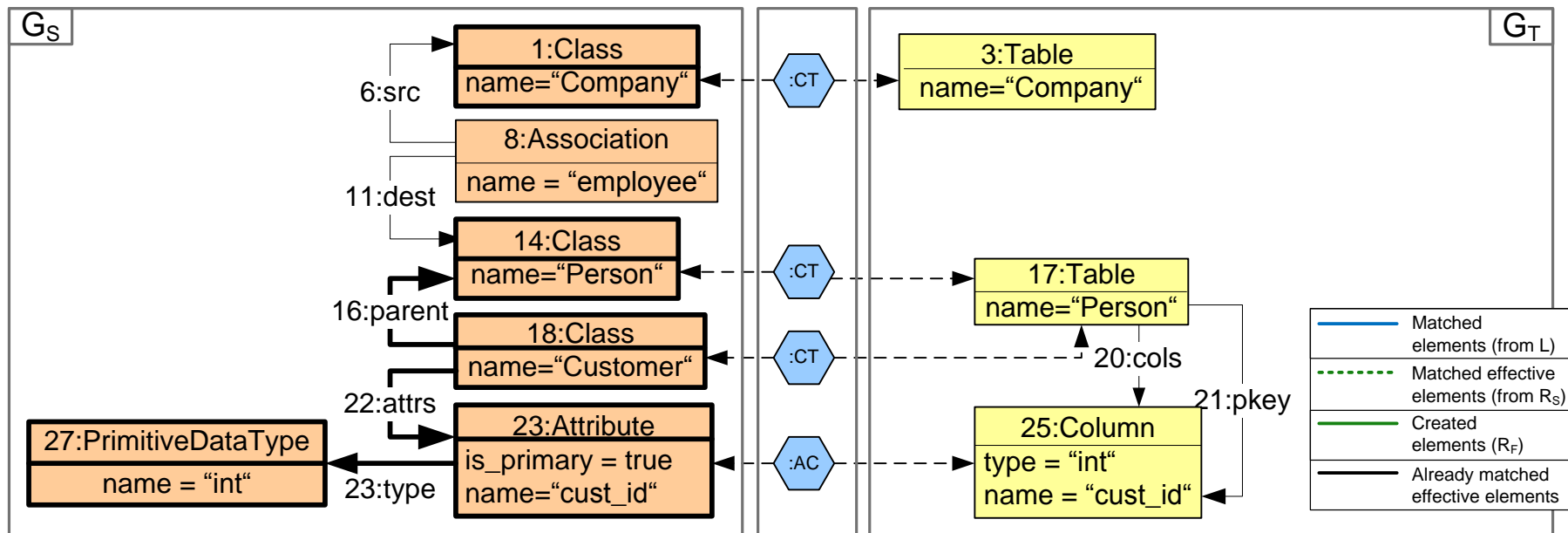
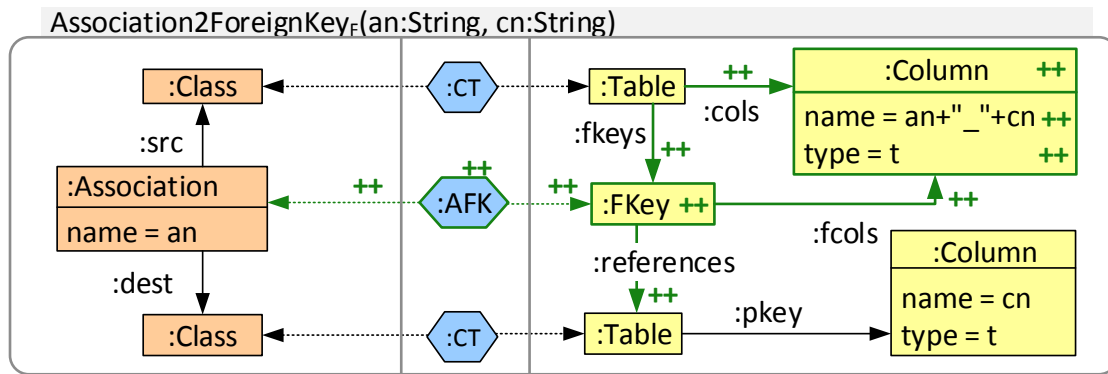
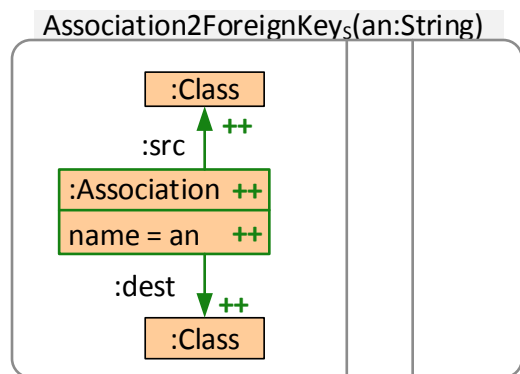
PrimaryAttr2Column_s(n:String, t:String)



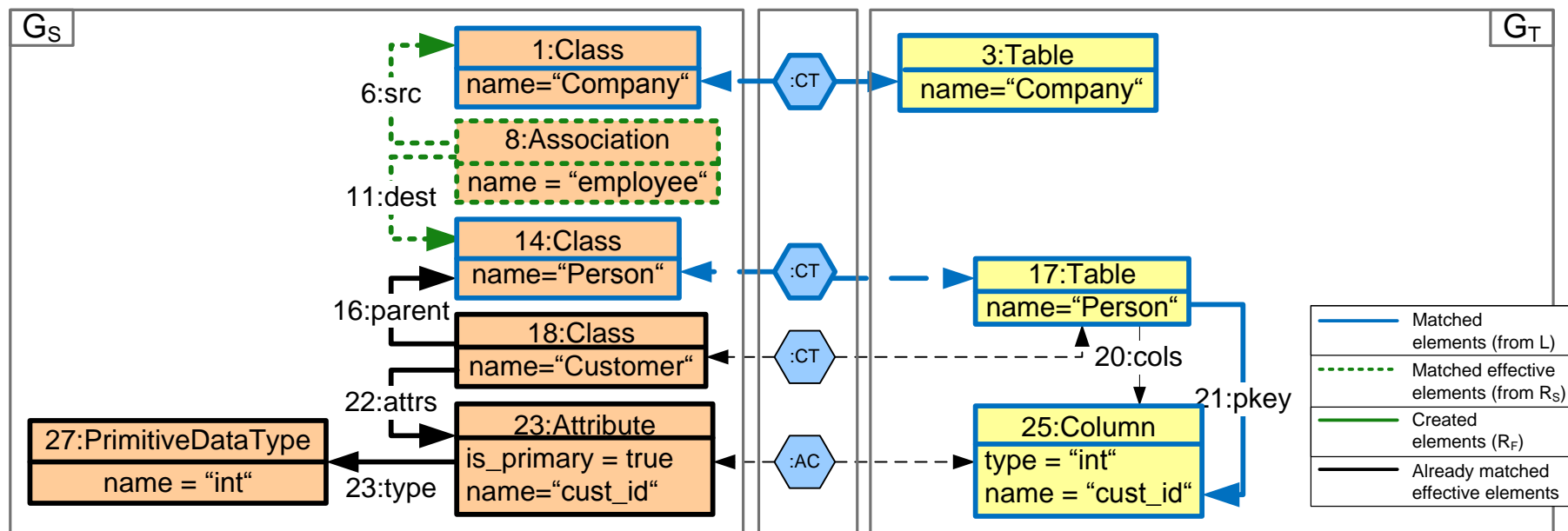
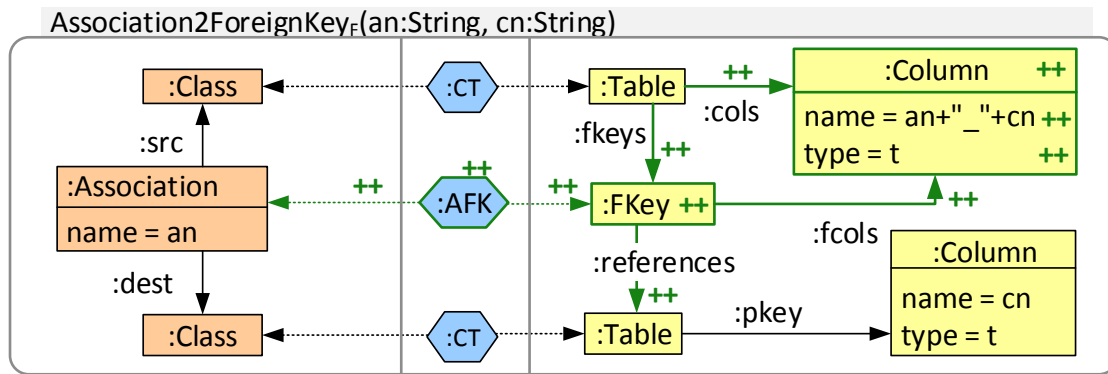
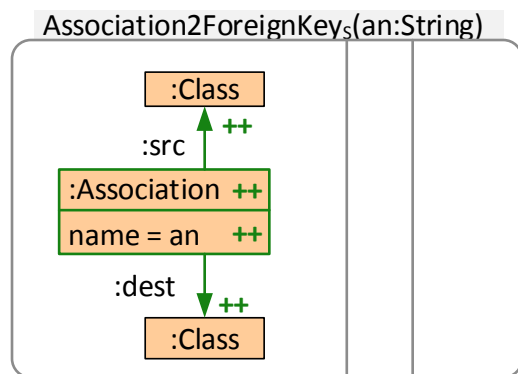
PrimaryAttr2Column_r(n:String, t:String)



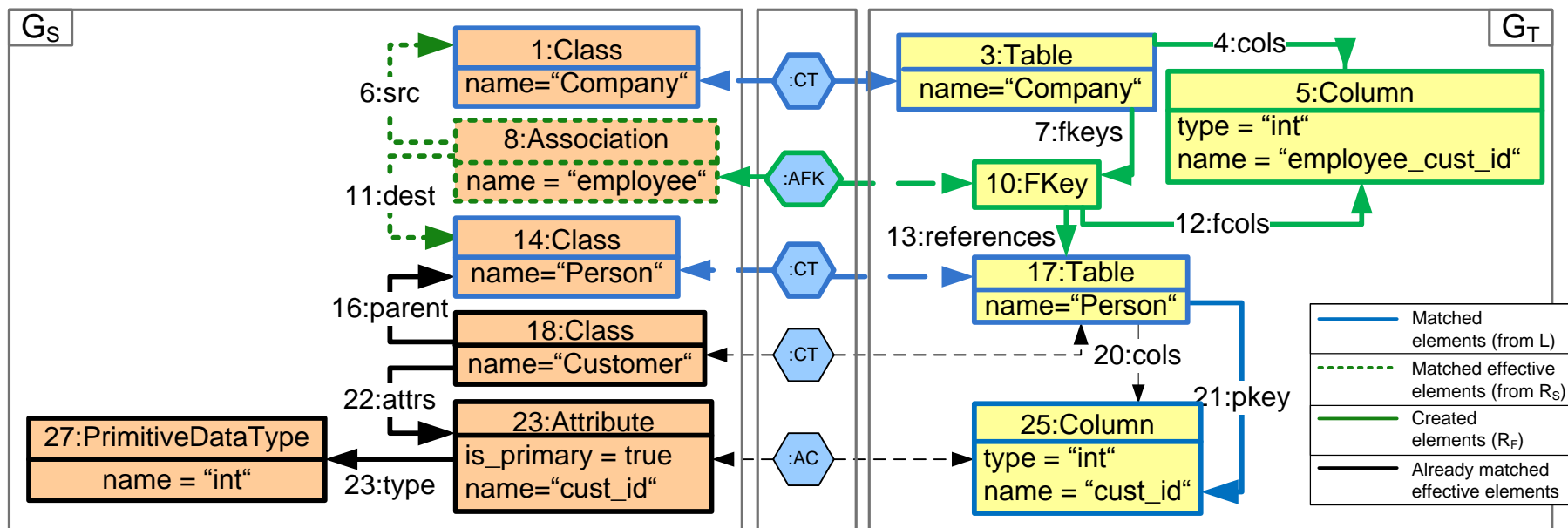
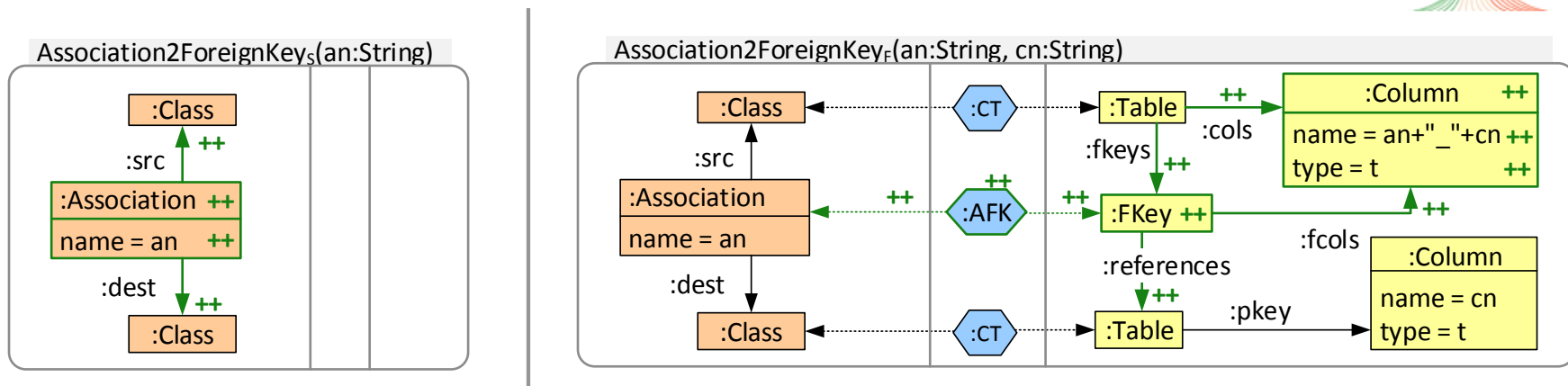
Example: Translation of Class Diagram



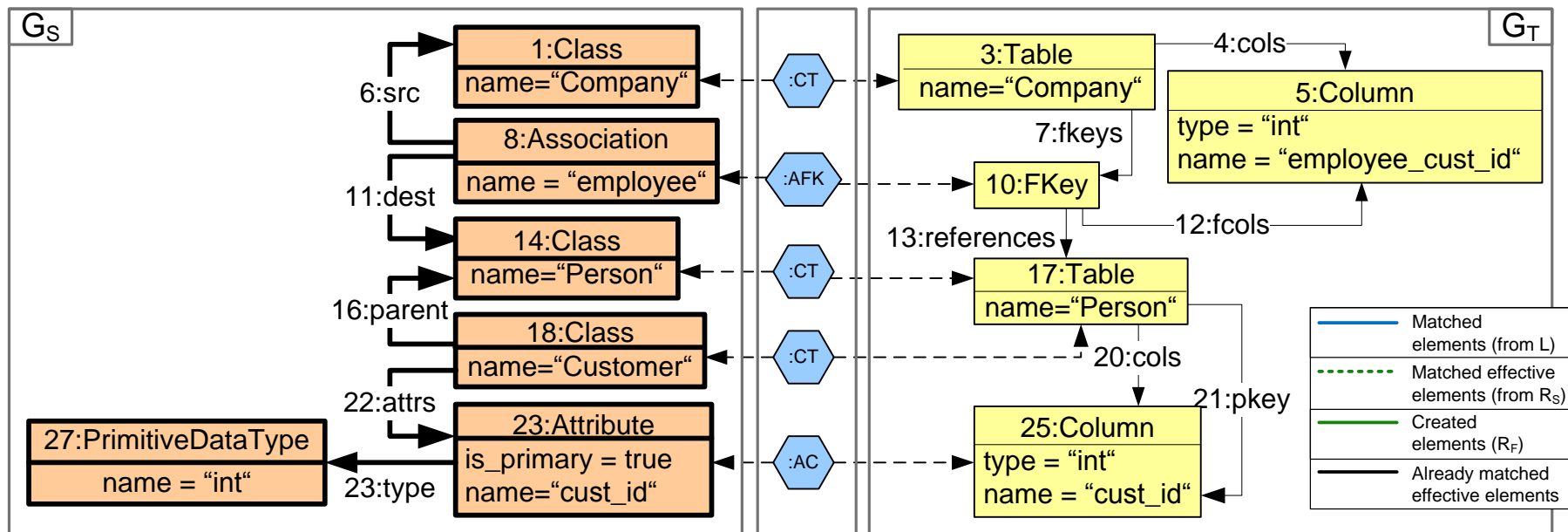
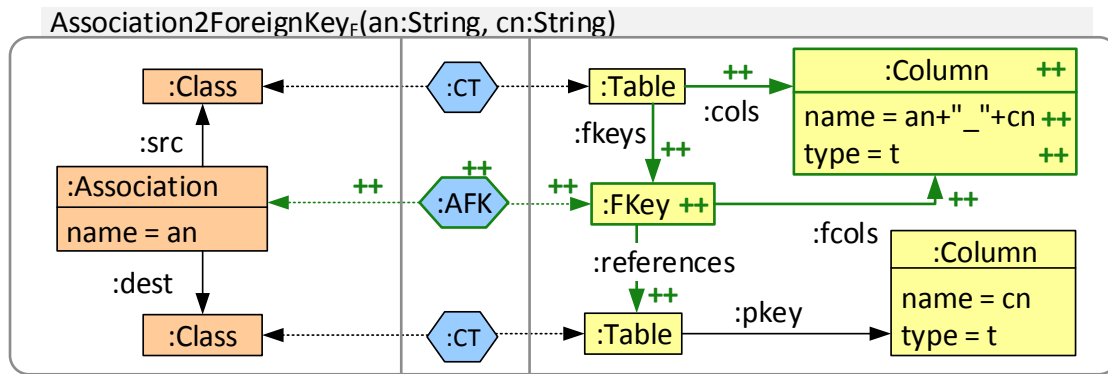
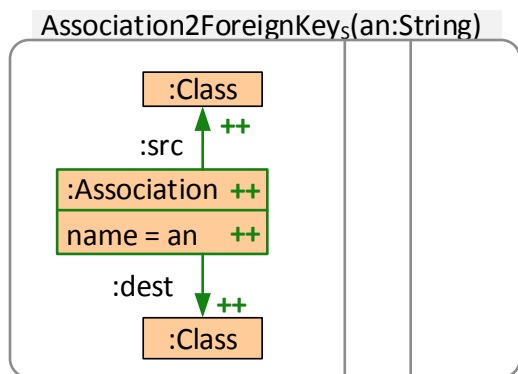
Example: Translation of Class Diagram



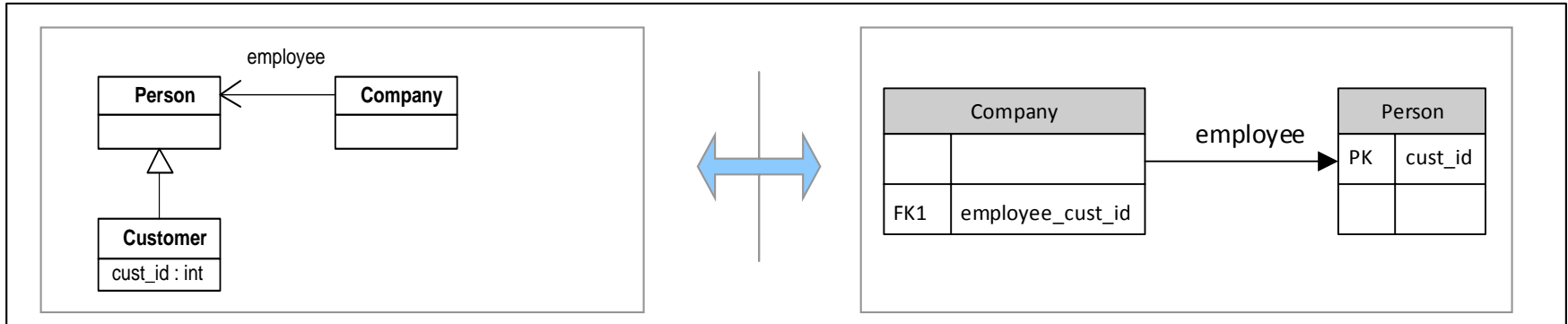
Example: Translation of Class Diagram



Example: Translation of Class Diagram

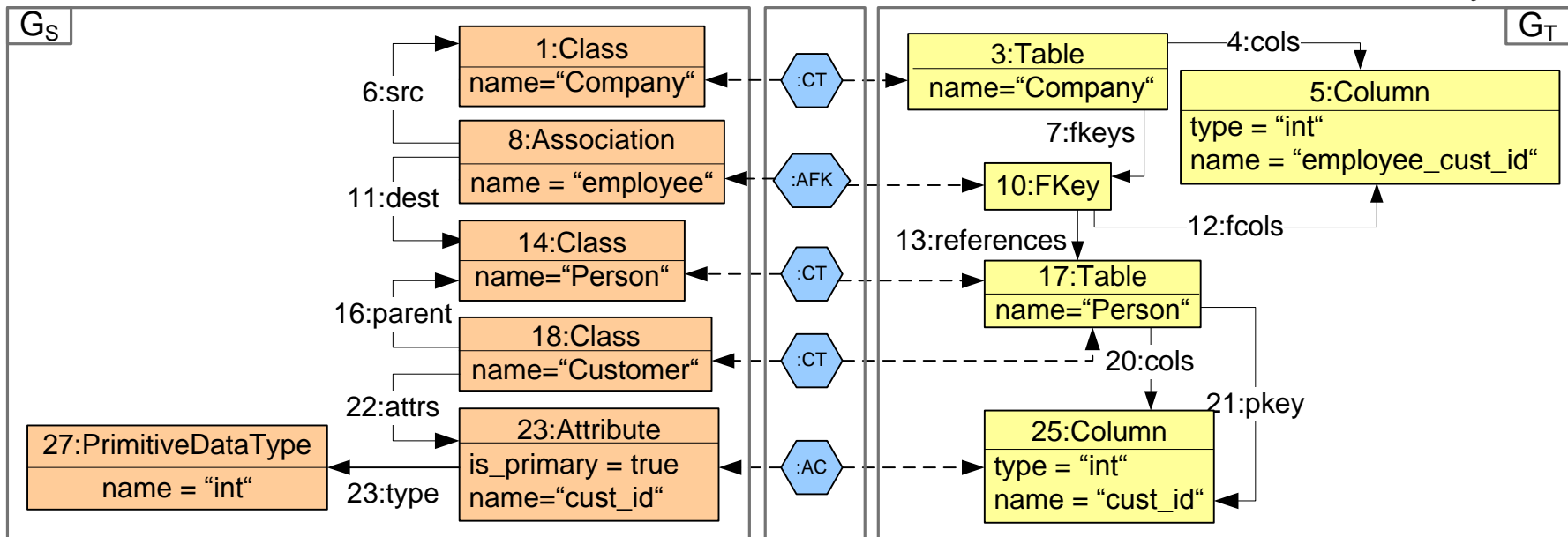


Example: Translation of Class Diagram

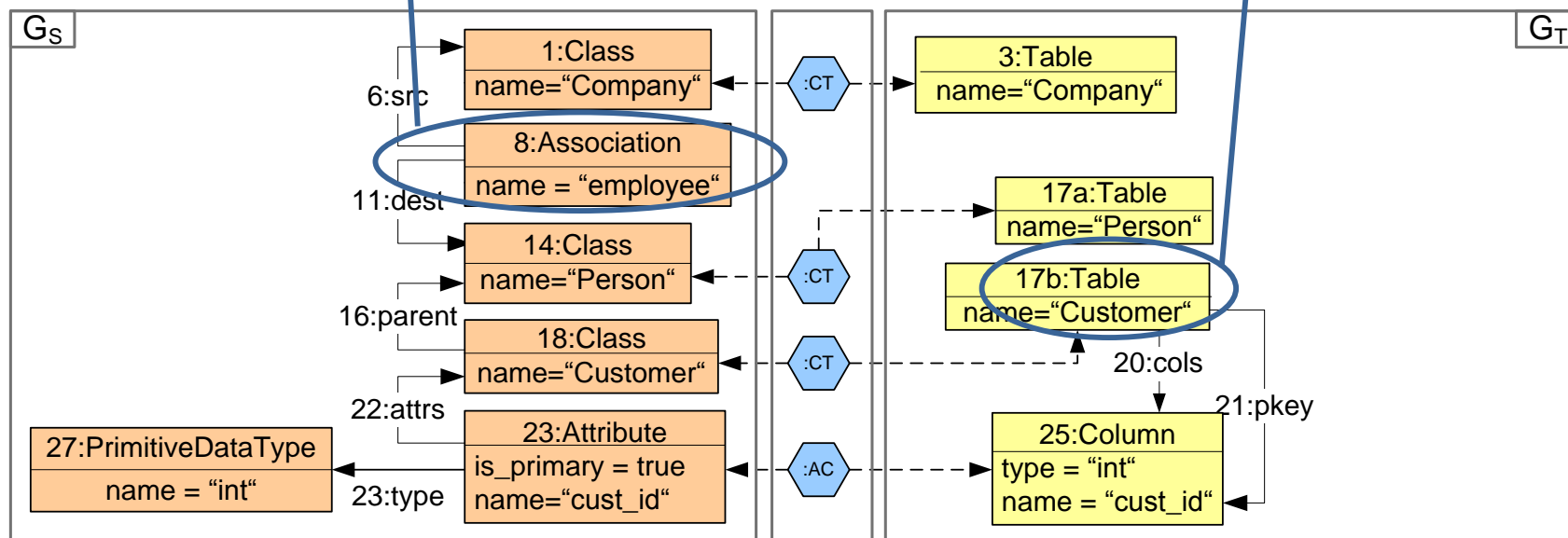
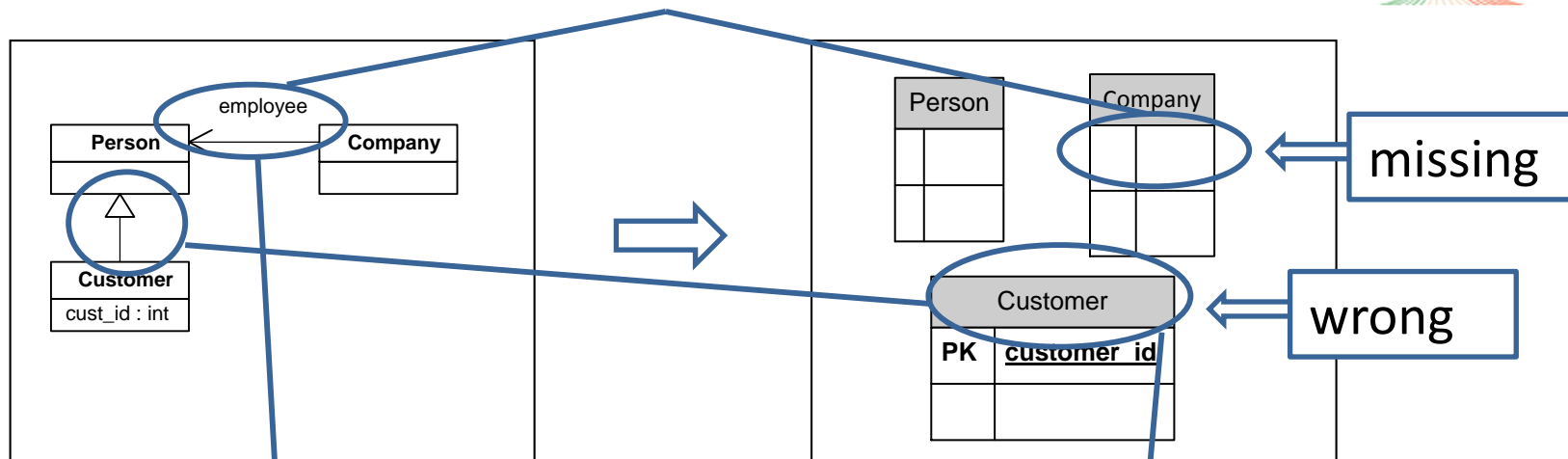


concrete syntax

abstract syntax



Example: inconsistent FW-MT

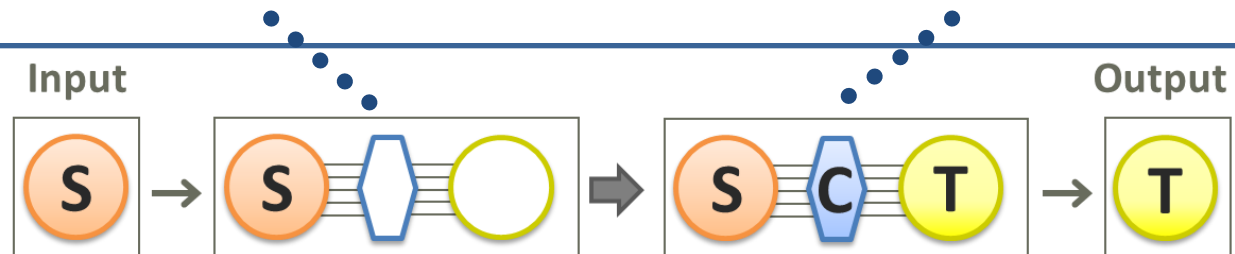


Definition (Model Transformation)

$MT : L(TG^S) \Rightarrow L(TG^T)$, is defined by **source consistent**

MT -sequences via $TR_F : (G^S, G_0 \xrightarrow{tr_F^*} G_n, G^T)$

where $G_0 = (G^S \leftarrow \emptyset \rightarrow \emptyset)$ and $G_n = (G^S \leftarrow G^C \rightarrow G^T)$



Theorem (Correctness, Completeness and Termination)

Each TGG-model transformation MT is

- **syntactically correct:** for each MT -sequence $(G^S, G_0 \xrightarrow{tr_F^*} G_n, G^T)$ there is $G = (G^S \leftarrow G^C \rightarrow G^T) \in L(TGG)$
- **complete:** for each source model $G^S \in L(TGG)^S$ there is an MT -sequence $(G^S, G_0 \xrightarrow{tr_F^*} G_n, G^T)$
- **terminating,** if all rules are source creating

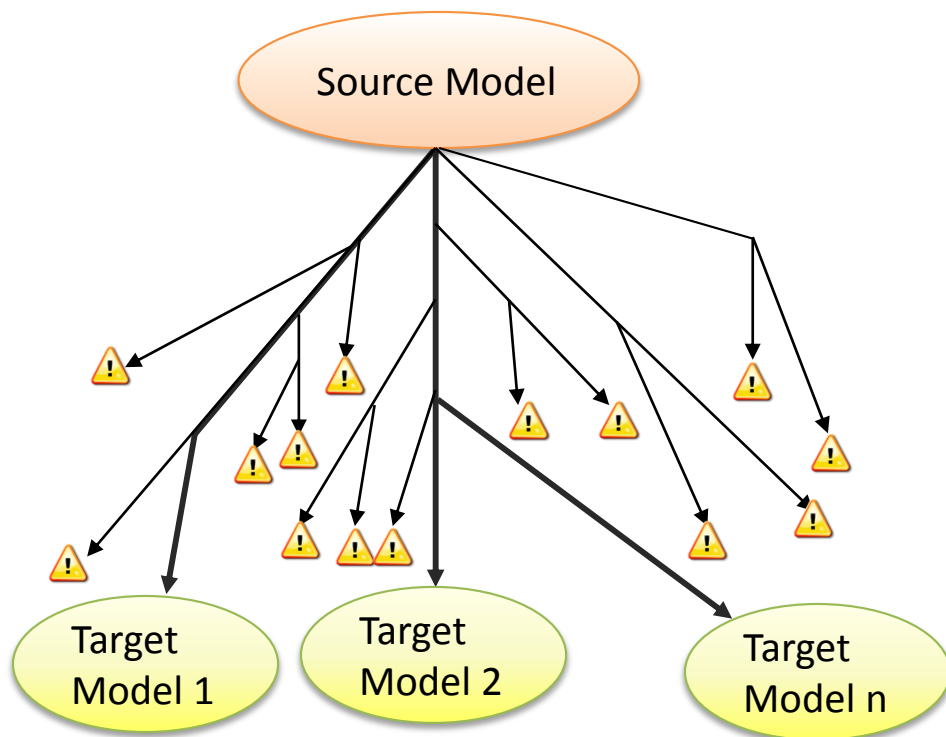
[EEE+07, EEHP09]

[EEE+07] H. Ehrig, K. Ehrig, C. Ermel, F. Hermann, and G. Taentzer: **Information Preserving Bidirectional Model Transformations**. *Proc. FASE'07*. Springer (2007).

[EEHP09] H. Ehrig, C. Ermel, F. Hermann, and U. Prange: **On-the-Fly Construction, Correctness and Completeness of Model Transformations based on Triple Graph Grammars**. *Proc. MODELS'09*. Springer (2009).

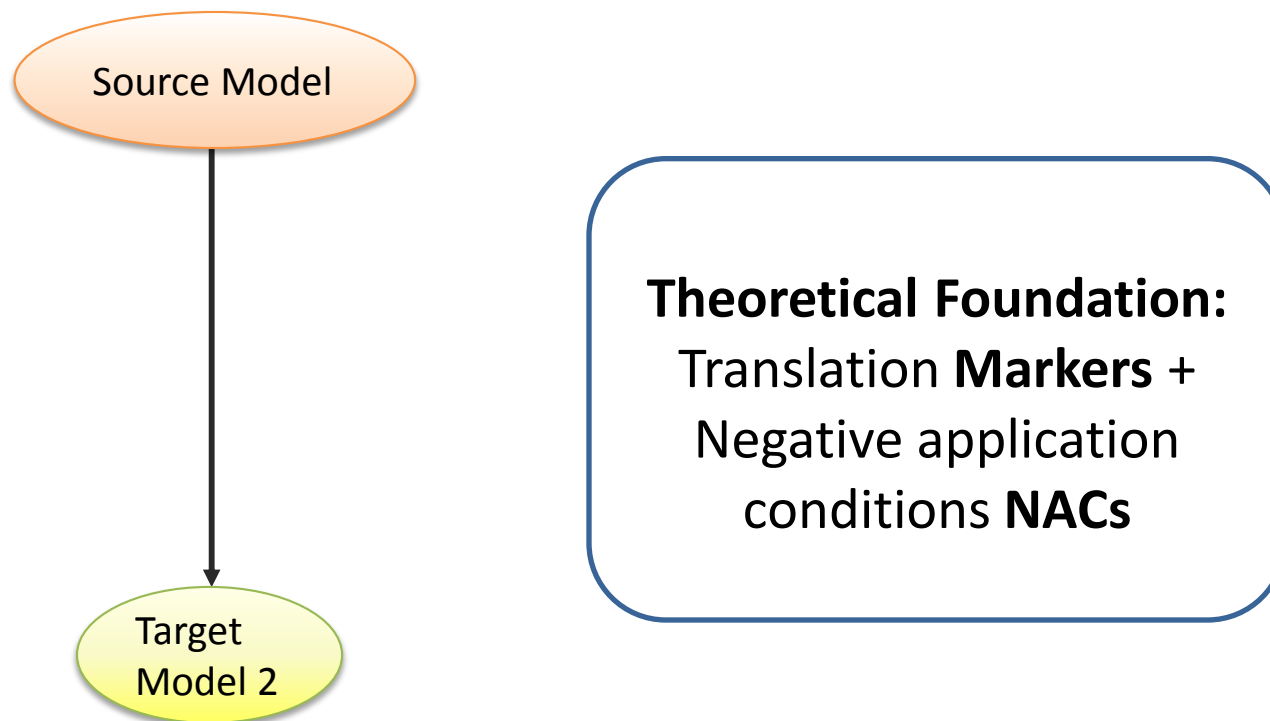
Aim: Avoid Backtracking

- **TASK:** restrict the applicability of translation rules, such that invalid paths are not constructed



Aim: Avoid Backtracking

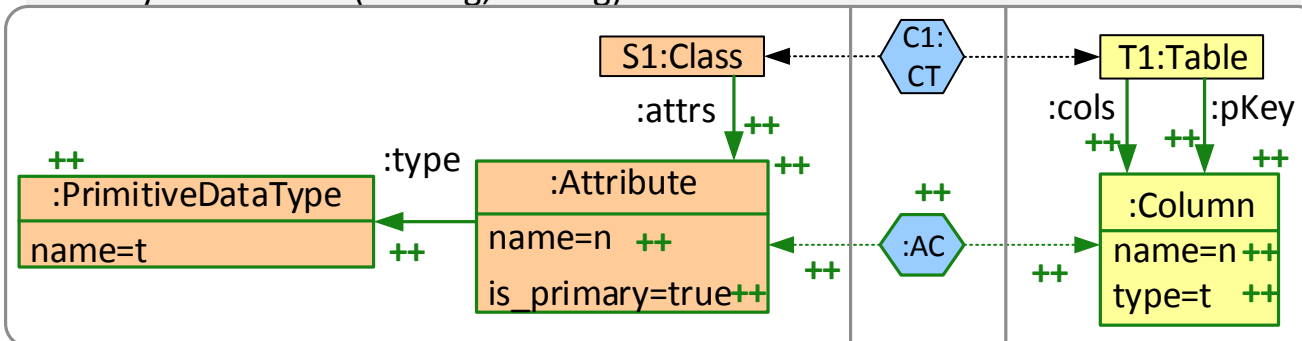
- **TASK:** restrict the applicability of translation rules, such that invalid paths are not constructed



Example: Translation Markers

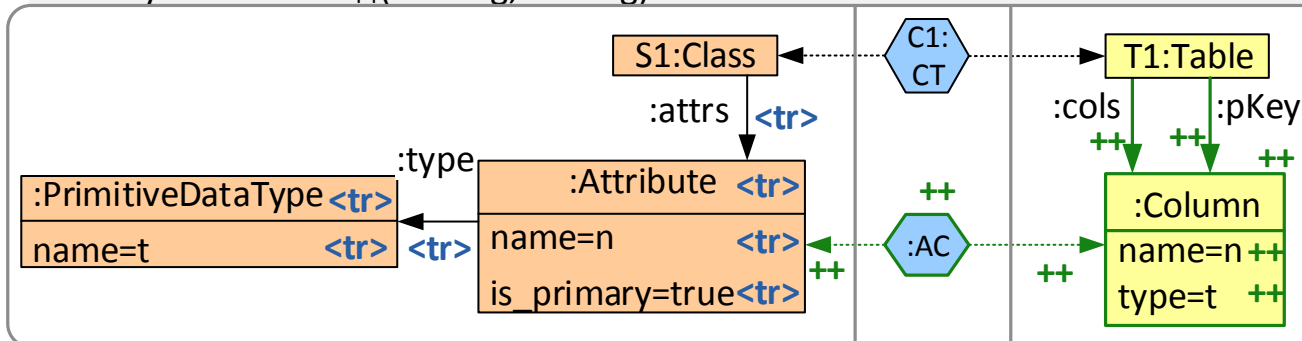
Triple Rule

PrimaryAttr2Column(n:String, t:String)



Forward Translation Rule

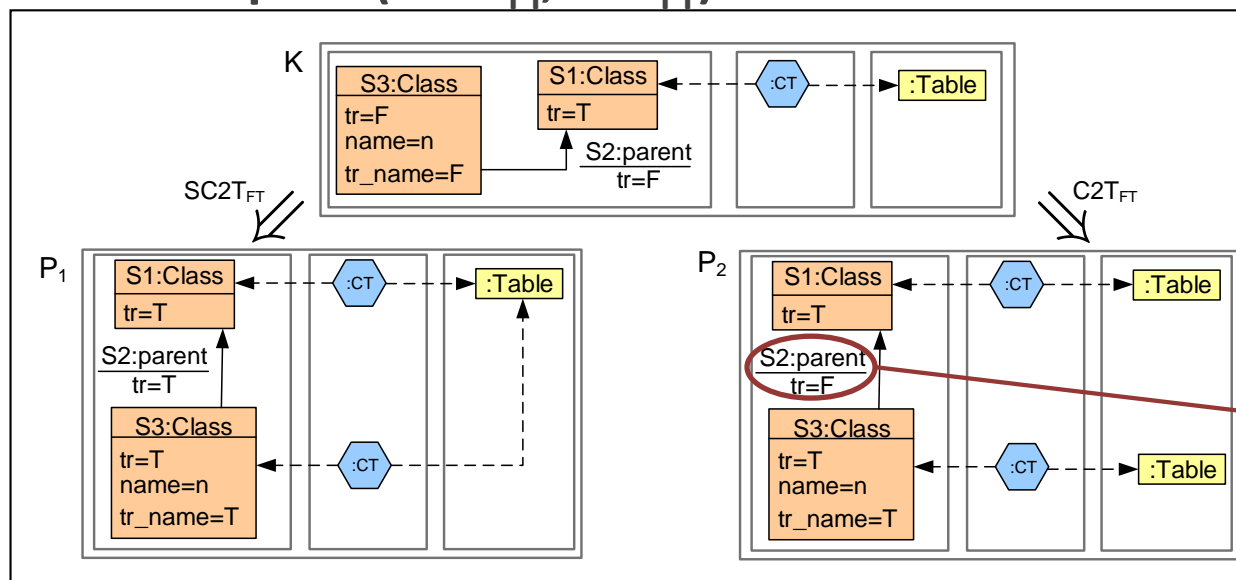
PrimaryAttr2Column_{FT}(n:String, t:String)



translation markers
<tr>

Conflicting Rules

- critical pair: ($SC2T_{FT}$, $C2T_{FT}$)



⚠ S2 remains untranslated [tr=F]

- $K \Rightarrow P_2$ via $C2T_{FT}$ leads to **backtracking** in any bigger context
- Tool support:** AGG for critical pair analysis
- Solution: K as **filter NAC** for $C2T_{FT}$

Theorem (Functional Behaviour + Efficiency)

Given: terminating MT via TR_{FT} .

If all **significant critical pairs** are **strictly confluent**

\Rightarrow

MT has **functional** behaviour and **does not require backtracking**.

[HEOG10] F. Hermann, H. Ehrig, F. Orejas, U. Golas: **Formal analysis of functional behaviour for model transformations based on triple graph grammars**. In: *Int. Conf. on Graph Transformations*, Springer (2010).

Tool support

eMoflon



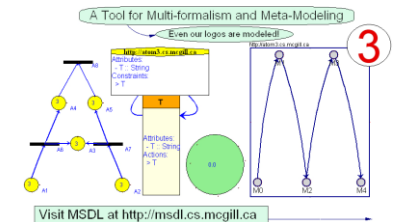
TU Darmstadt (D)

HenshinTGG



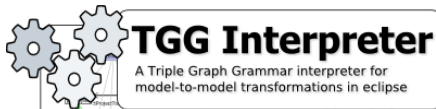
TU Berlin (D),
Uni Luxembourg (LU),
Uni Marburg (D)

ATOM³



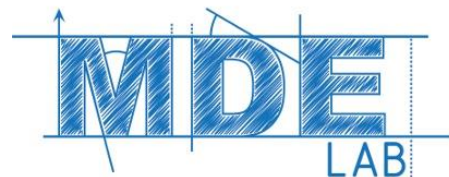
Uni Madrid (ES),
McGill University (CA)

TGG Interpreter



Uni Paderborn (D)

MDE LAB



HPI – Potsdam (D)

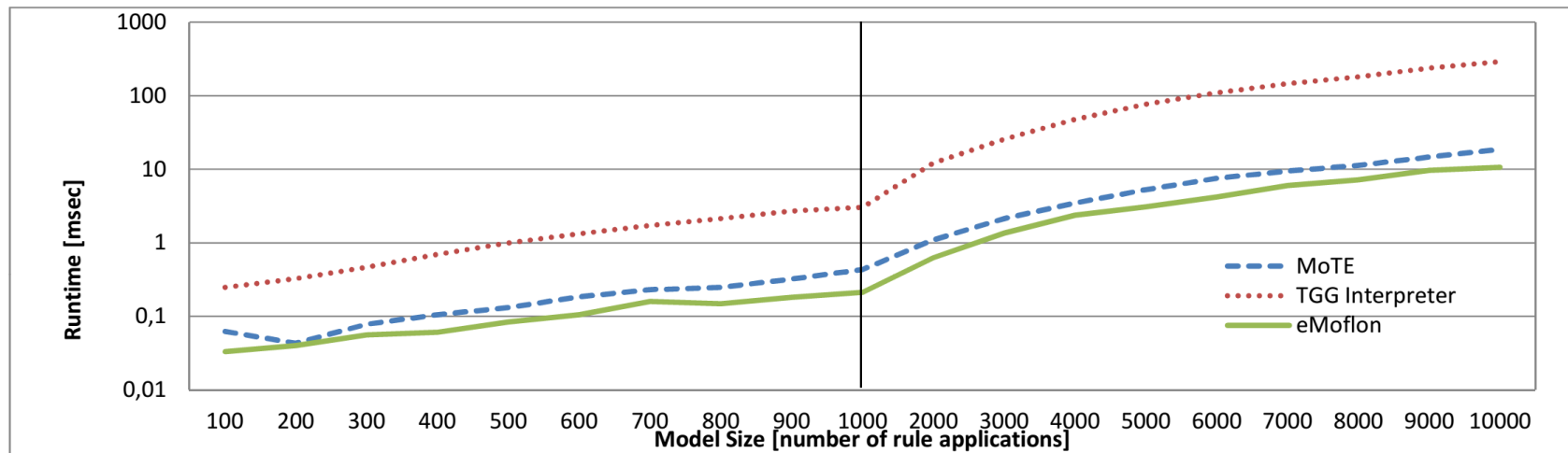
EMorF



Solunar GmbH –
Gütersloh (D)

Benchmark

- Example: CD2RDBM (without NACs)
- Synthetic benchmark – instances are generated



[HLG+13] S. Hildebrandt, L. Lambers, H. Giese, J. Rieke, J. Greenyer, W. Schäfer, M. Lauder, A. Anjorin, A. Schürr: **A Survey of Triple Graph Grammar Tools.** *Preproceedings BX 2013.*

Application of TGGs in the large at SES

Scenario: Translation of satellite control procedures

Input size: 400 KLOC; 1.5 mio. nodes (4,000 per file)

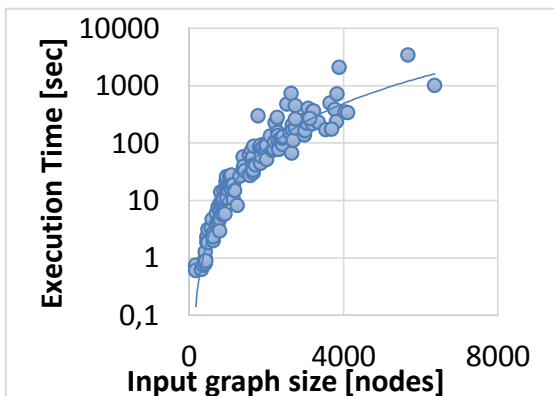
TGG: 249 rules + 235 initialization/refactoring rules

Performance Improvement

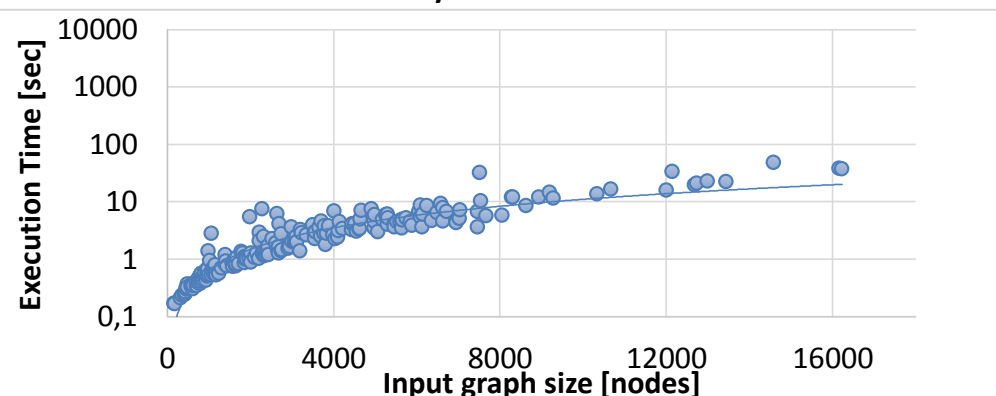
- **Filter NACs**
+ **DC-layering** (order on minimal groups of rules)



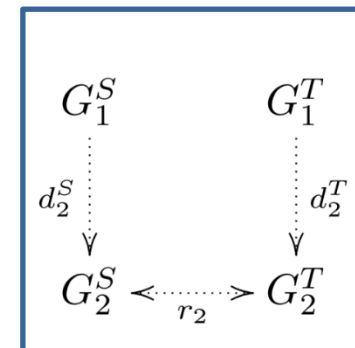
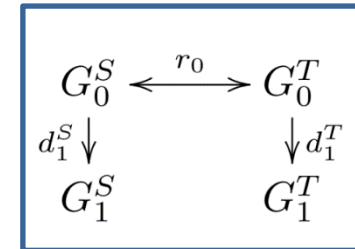
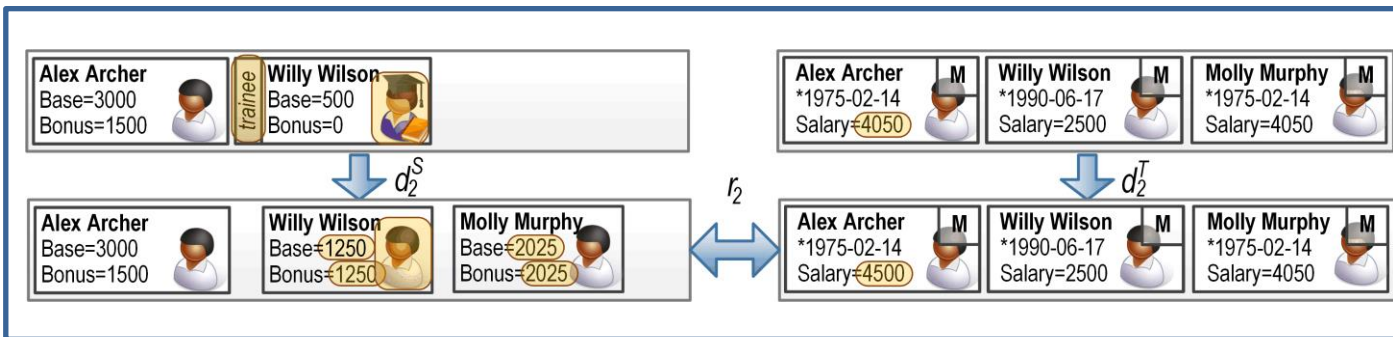
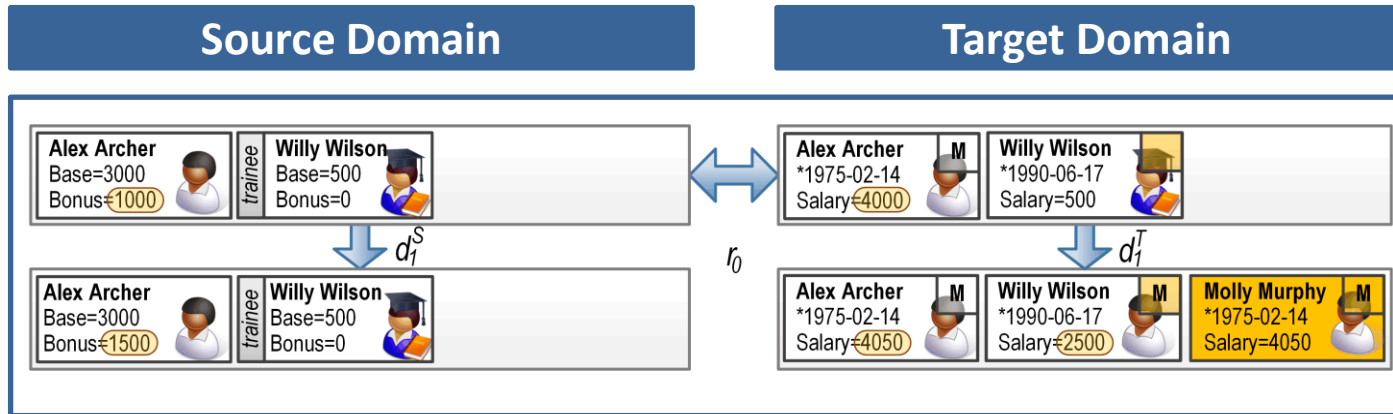
Pure TGG execution



DC-layered TGG

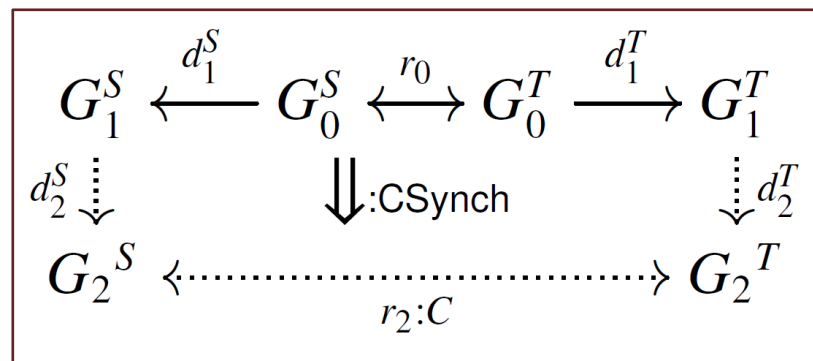


Concurrent Synchronization Problem

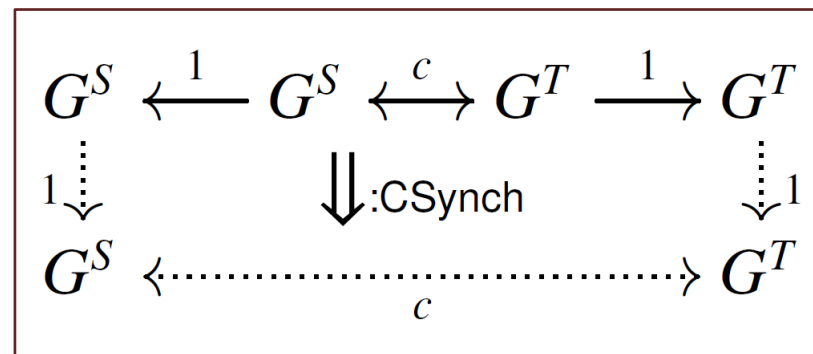


Correctness

- Consistency Law:**
result is always consistent

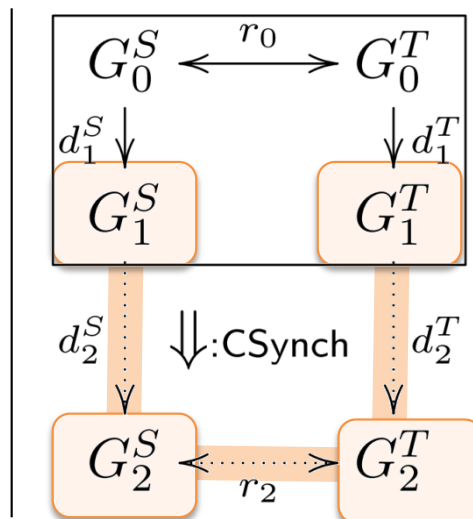
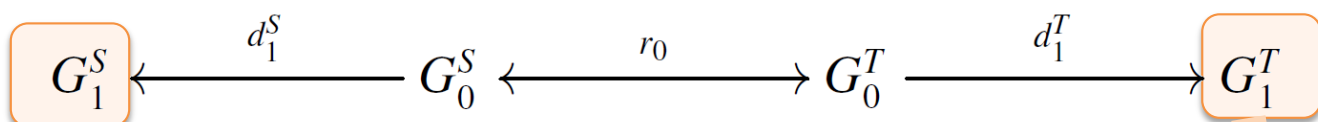


- Identity Law:**
no change, if input is already consistent



Concurrent Synchronization: General Concept

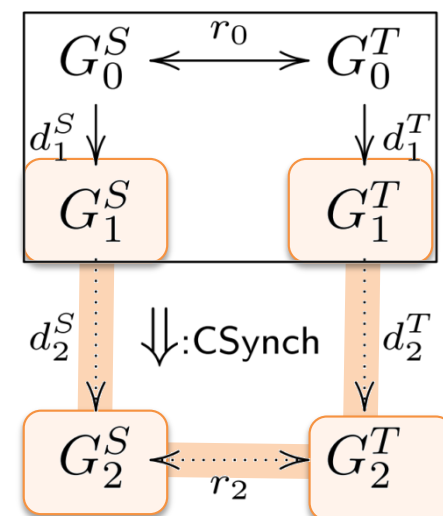
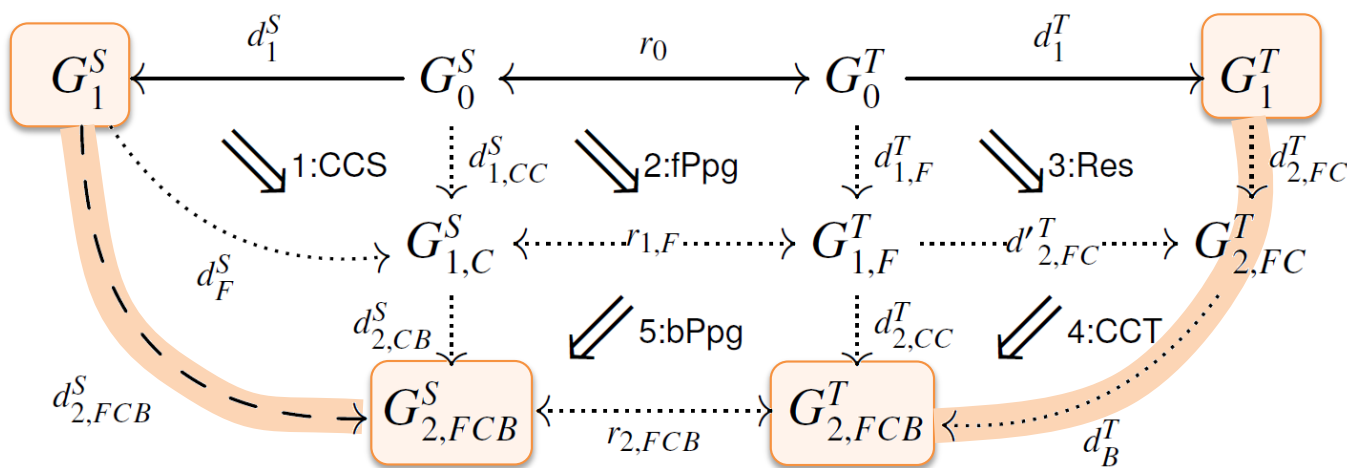
- **1:CCS** = Consistency Checking on Source component
- **2:fPpg** = FW-Propagation
- **3:Res** = Conflict Resolution
- **4:CCT** = Consistency Checking on Target component
- **5:bPpg** = BW-Propagation



[HEEO12] H. Ehrig, C. Ermel, G. Taentzer: Concurrent Model Synchronization with Conflict Resolution Based on Triple Graph Grammars. In: Proc. FASE'12. Springer (2012).

Concurrent Synchronization: General Concept

- **1:CCS** = Consistency Checking on Source component
- **2:fPpg** = FW-Propagation
- **3:Res** = Conflict Resolution
- **4:CCT** = Consistency Checking on Target component
- **5:bPpg** = BW-Propagation



[HEEO12] H. Ehrig, C. Ermel, G. Taentzer: Concurrent Model Synchronization with Conflict Resolution Based on Triple Graph Grammars. In: Proc. FASE'12. Springer (2012).

Result: Correctness and Completeness



Definition (Completeness)

Model synchronisation can be performed for any input.

Theorem (Correctness and Completeness)

Given a triple graph grammar TGG, the derived nondeterministic concurrent synchronisation framework $\text{CSync}(\text{TGG}, \text{CSync})$ is **correct** and **complete**.

[GHN+13] S. Gottmann, F. Hermann, N. Nachtigall, B. Braatz, C. Ermel, H. Ehrig, T. Engel: Correctness and Completeness of Generalised Concurrent Model Synchronisation Based on Triple Graph Grammars. In: Proc. AMT'13, CEUR 2013.

[HEEO12] H. Ehrig, C. Ermel, G. Taentzer: Concurrent Model Synchronization with Conflict Resolution Based on Triple Graph Grammars. In: Proc. FASE'12. Springer (2012).

Cross-disciplinary concepts

- **Different strategies:**
 - Batch / incremental execution
 - Deterministic / non-deterministic execution
 - Single / concurrent updates

- **Relationship between TGGs and lenses:**
 - View TGGs [ARDS13]
 - Delta Lenses and TGGs [HEO+13]

[ARDS13] Anjorin, A., Rose, S., Deckwerth, F., Schürr, A.: Asymmetric delta lenses with view triple graph grammars (to appear). ECEASST (2013)

[HEO+13] F Hermann, H Ehrig, F Orejas, K Czarnecki, Z Diskin, Y Xiong, S Gottmann, T Engel: Model synchronization based on triple graph grammars: correctness, completeness and invertibility. In: Software & Systems Modeling, 1-29, Springer 2013.

Possible impact from other BX communities

Concepts and Methods for:

- Information preservation (DB, ...)
- Data protection (DB, SE, ...)
- Efficient updates, large scale systems (DB, SE, ...)
- Exception handling (PL, SE, ...)
- General framework via lenses (PL, SE, ...)
- ...

General results:

- Lenses properties (PL, SE, ...)
- ...

Summary

- **Formal Guarantees:**
syntactical correctness, completeness, termination,
- **Formal Analysis:**
functional behavior, information preservation, invertibility,
semantical correctness
- **General concept:**
for concurrent model synchronization
- **Tool** support and **industrial** use

Further Reading



[EEE+07]	H. Ehrig, K. Ehrig, C. Ermel, F. Hermann, and G. Taentzer: Information Preserving Bidirectional Model Transformations . <i>Proc. FASE'07</i> . Springer (2007).
[EEPT06]	H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer: Fundamentals of Algebraic Graph Transformation . EATCS Monographs in Theoretical Computer Science. Springer (2006).
[GHL12]	Giese, H., Hildebrandt, S., Lambers, L.: Bridging the Gap Between Formal Semantics and Implementation of Triple Graph Grammars. Ensuring Conformance of Relational Model Transformation Specifications and Implementations . <i>Software and Systems Modeling</i> , Springer (2012).
[GW09]	Giese, H., Wagner, R.: From model transformation to incremental bidirectional model synchronization . <i>Software and Systems Modeling</i> 8(1), Springer (2009).
[HEGO10]	F. Hermann, H. Ehrig, U. Golas, Fernando Orejas: Efficient Analysis and Execution of Correct and Complete Model Transformations Based on Triple Graph Grammars . <i>Proc. of MDI'10</i> , ACM (2010).
[HEOG10]	F. Hermann, H. Ehrig, F. Orejas, U. Golas: Formal analysis of functional behaviour for model transformations based on triple graph grammars . In: <i>Int. Conf. on Graph Transformations</i> , Springer (2010).

Further Reading



[HEO+13]	F. Hermann, H. Ehrig, F. Orejas, K. Czarnecki, Z. Diskin, Y. Xiong, S. Gottmann, T. Engel: Model synchronization based on triple graph grammars: correctness, completeness and invertibility. In: <i>Software & Systems Modeling</i> , Springer 2013.
[HEGO14]	F. Hermann, H. Ehrig, U. Golas, F. Orejas: Formal Analysis of Model Transformations Based on Triple Graph Grammars. In <i>MSCS</i> , to appear 2014.
[KW07]	Kindler, E., Wagner, R.: Triple graph grammars. concepts, extensions, implementations, and application scenarios. <i>Tech. Rep. TR-ri-07-284</i> , Department of Computer Science, University of Paderborn (2007).
[LAVS12]	Lauder, M., Anjorin, A., Varró, G., Schürr, A.: Bidirectional model transformation with precedence triple graph grammars. In: <i>Proc. Eur. Conf. on Modelling Foundations and Applications (ECMFA'12)</i> , LNCS, vol. 7349. Springer (2012).
[Schürr94]	Schürr, A.: Specication of Graph Translators with Triple Graph Grammars. <i>Proc. of WG 1994</i> . LNCS, Springer (1995).
[SK08]	Schürr, A., Klar, F.: 15 years of triple graph grammars. In: <i>Int. Conf. on Graph Transformations (ICGT 2008)</i> . LNCS, vol. 5214, Springer (2008).