Strategic Term Rewriting and Its Application to a VDM-SL to SQL Conversion

Review

Outline

- Goal of the paper
- Algebraic design by calculation
- VooDooM model
- Conclusions and future work

Goal of the paper

Convert datatypes in VDM-SL
to
SQL relational data models

How?

- **Conversion**: Transforming algebraic types to maps and products
- **Implementation technology**: Haskell

Database design by calculation

Abstraction and representation

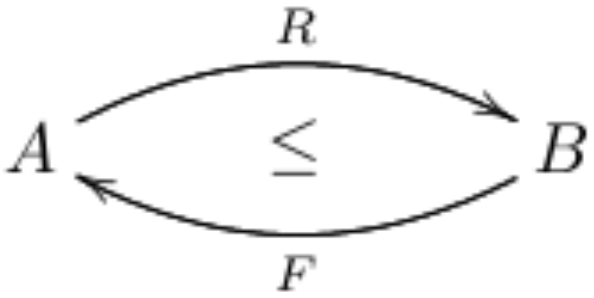
$A \xleftarrow{F} B$ (the *abstraction relation*)

$A \xrightarrow{R} B$ (the *representation relation*)

where:

$$F \cdot R = id_A$$

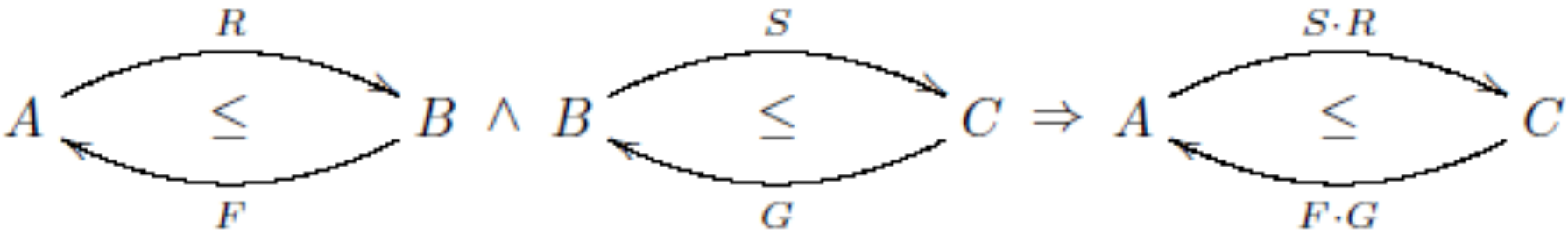
Database design by calculation



Datatype B implements or refines datatype A

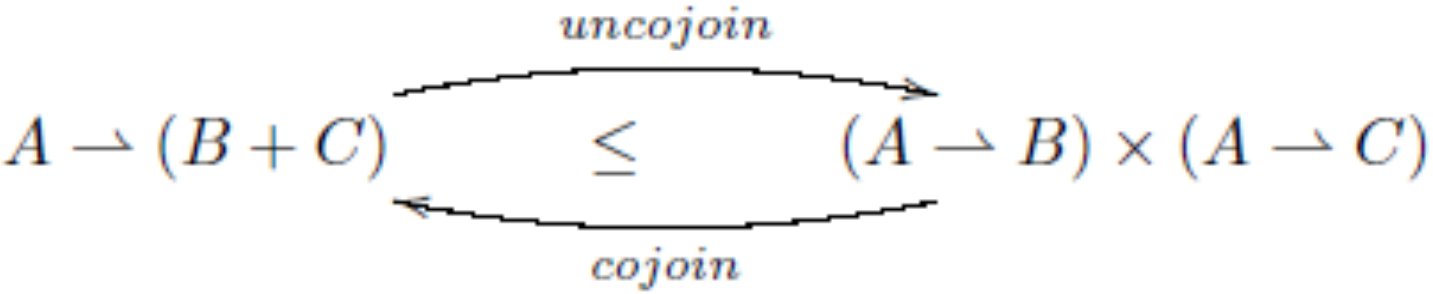
Database design by calculation

Preorder



Database design by calculation

Conversion laws

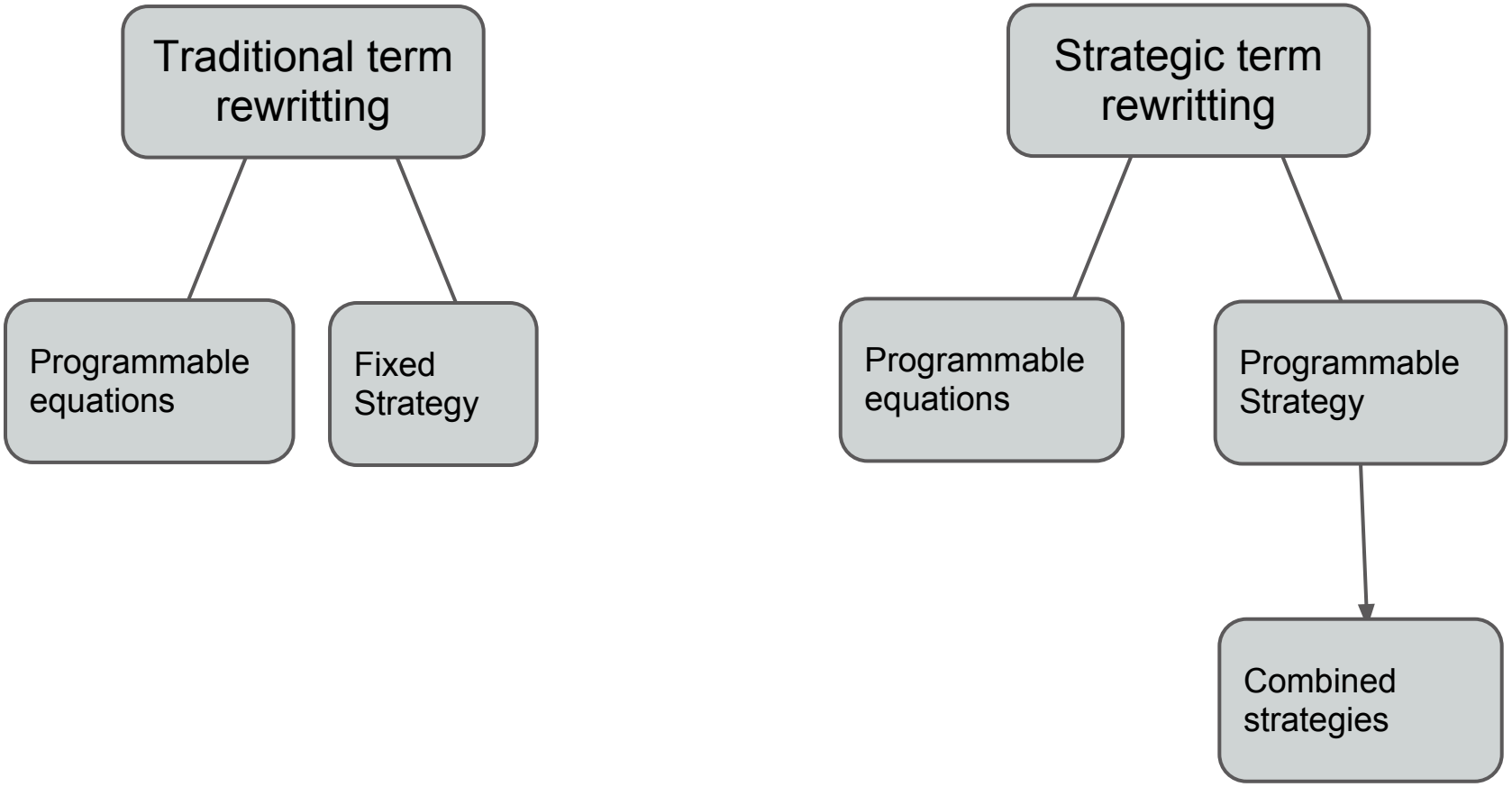


Database design by calculation

Normal form

$$DB = \prod_{i=1}^{n} \left(\prod_{j=0}^{n_i} K_j \rightharpoonup \prod_{k=0}^{m_i} D_k \right)$$

Strategic term rewriting



VooDooM Tool

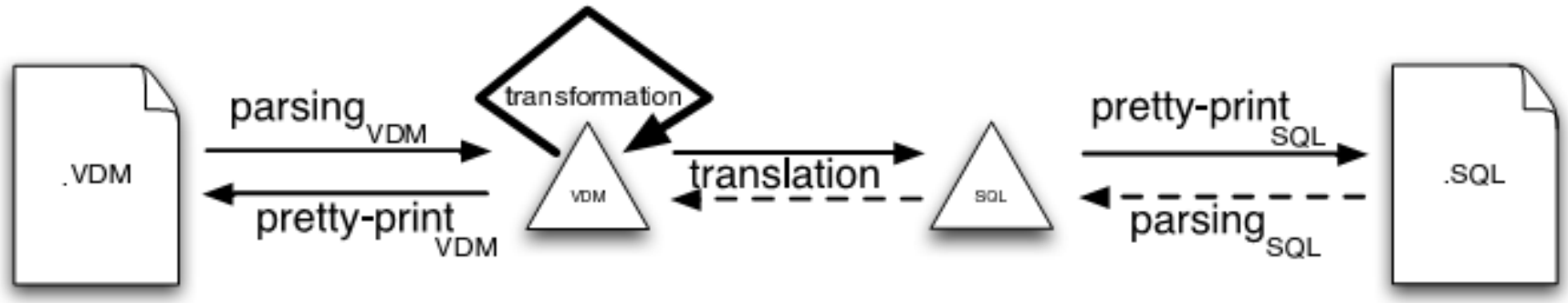


Fig. 2. Overall architecture of the VooDooM tool

Transformation

Refinement of the VDM-SL data types to a relational form

- Inlining and recursion removal
- Desugaring
- Conversion to relational form
- Resugaring

SQL translation

Table 2. Correspondence between VDM-SL and SQL92 data types

VDM-SL data type	SQL data type	SQL Constraint
bool	SMALLINT	CHECK (.. IN (0,1))
nat	INT	CHECK .. >= 0
nat1	INT	CHECK .. >= 1
int	INT	
rat	REAL	
real	REAL	
char	CHAR (1)	
token	VARCHAR (128)	
seq of char	VARCHAR (128)	

Conclusions and future work

- Automatic database schema generation
- In comparison with other approaches with the VooDooM the source data-model can be arbitrarily complex
- Reverse process: obtain algebraic data types from a relational model