

Reaping the Benefits of Modularization in Flexiformal Mathematics by GF-based AST Transformations

Josefin Kelber², Michael Kohlhase¹, Jan Frederik Schaefer¹, Marcel Schütz¹

[1] FAU Erlangen-Nürnberg, [2] Technical University of Munich

07 October 2025

CICM 2025, Brasilia, Brazil, 6–10 October 2025

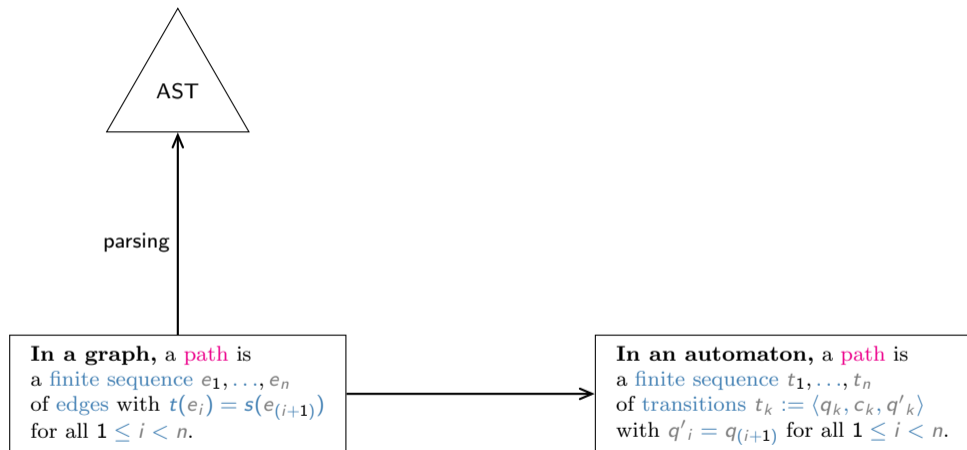


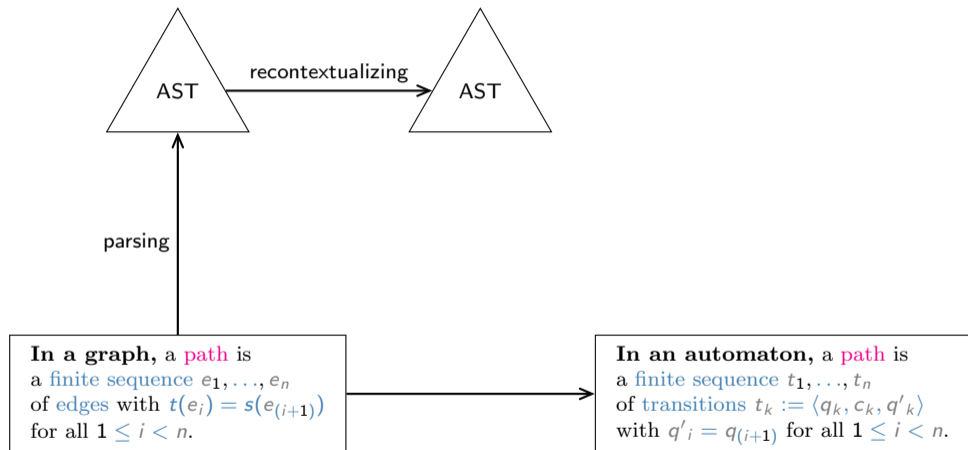
Overview

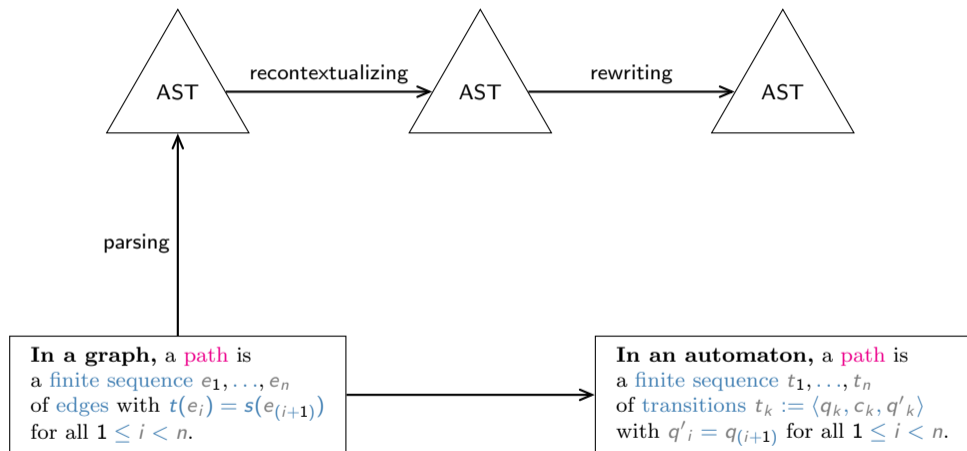
In a graph, a **path** is a finite sequence e_1, \dots, e_n of edges with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.

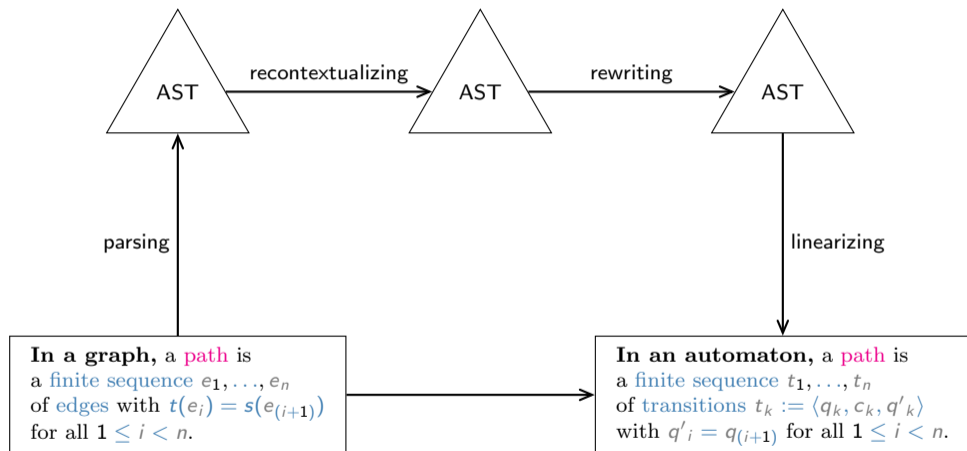


In an automaton, a **path** is a finite sequence t_1, \dots, t_n of transitions $t_k := \langle q_k, c_k, q'_k \rangle$ with $q'_i = q_{i+1}$ for all $1 \leq i < n$.









Motivation

University Education:

- ▶ many students with diverse backgrounds
 - ▶ few teachers
- ⇒ difficult to provide individualized learning material

University Education:

- ▶ many students with diverse backgrounds
 - ▶ few teachers
- ⇒ difficult to provide individualized learning material

Adaptive Learning Assistants:

- ▶ can track student's learning levels
 - ▶ can adapt learning material accordingly
- ⇒ can complement teachers

Motivation: The ALEA System



<https://courses.voll-ki.fau.de>

Motivation: The ALEA System



<https://courses.voll-ki.fau.de>

reachable

Definition 0.0. A **path** is a **finite sequence** e_1, \dots, e_n of **edges** e_i with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.

Definition 0.0. A **state** x is said to be **reachable** from a **state** y iff there exists a **path** from y to x .

Motivation: The ALEA System



<https://courses.voll-ki.fau.de>

reachable

Definition 0.0. A path is a finite sequence e_1, \dots, e_n of edges e_i with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.

Definition 0.0. A state x is said to be reachable from a state y iff there exists a path from y to x .

reachable

Definition 0.0. A path is a finite sequence t_1, \dots, t_n of transitions $t_i := (q_i, c_i, q'_i)$ such that $q'_i = q_{i+1}$ for all $1 \leq i < n$.

Definition 0.0. A state x is said to be reachable from a state y iff there exists a path from y to x .

Recontextualization

Recontextualization: Translation Along Theory Morphisms

Labelled Graph

V	(nodes)	:	set
E	(edges)	:	set
L	(labels)	:	set
s	(source)	:	$E \rightarrow V$
t	(target)	:	$E \rightarrow V$
l	(label)	:	$E \rightarrow V$

Recontextualization: Translation Along Theory Morphisms

Labelled Graph

V	(nodes)	:	set
E	(edges)	:	set
L	(labels)	:	set
s	(source)	:	$E \rightarrow V$
t	(target)	:	$E \rightarrow V$
l	(label)	:	$E \rightarrow V$

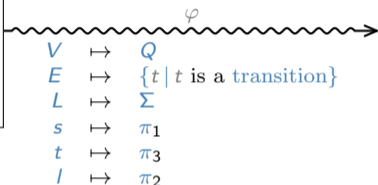
Automaton

Q	(states)	:	set
Σ	(symbols)	:	set
δ	(trans. rel.)	:	$Q \times \Sigma \rightarrow \mathcal{P}(Q)$
q_0	(init. state)	:	Q
F	(fin. states)	:	$\mathcal{P}(Q)$

A **transition** is a triple $\langle q, c, q' \rangle$
with $q' \in \delta(q, c)$.

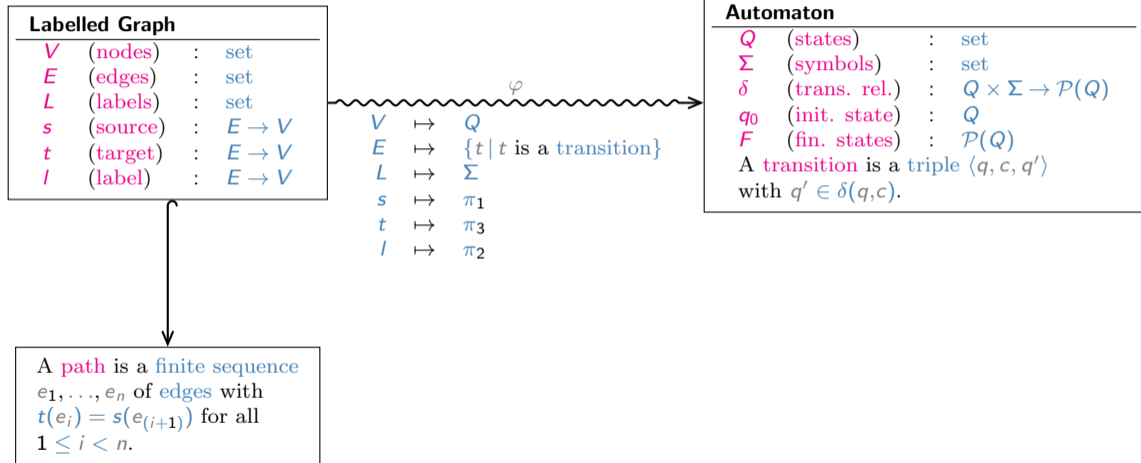
Recontextualization: Translation Along Theory Morphisms

Labelled Graph		
V	(nodes)	: set
E	(edges)	: set
L	(labels)	: set
s	(source)	: $E \rightarrow V$
t	(target)	: $E \rightarrow V$
l	(label)	: $E \rightarrow V$

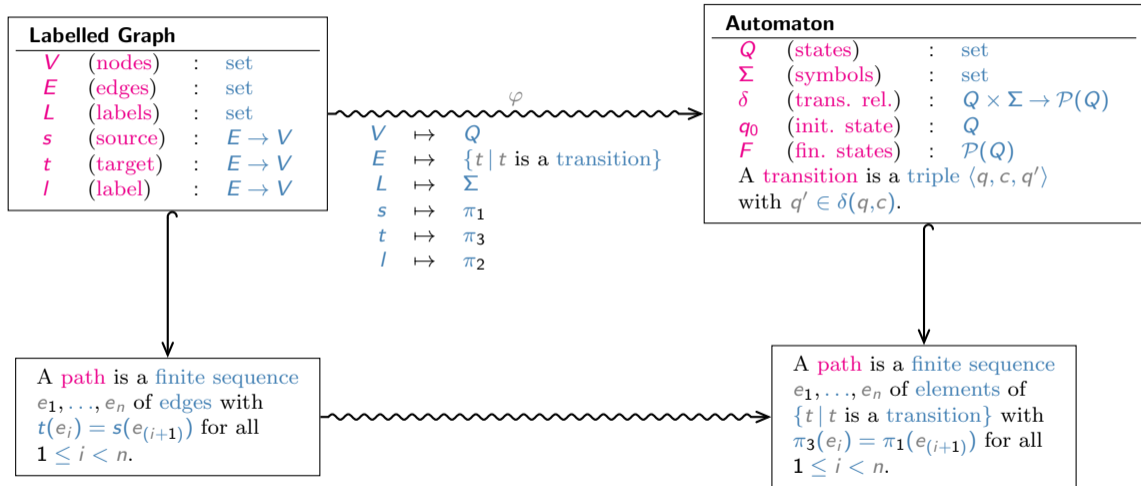


Automaton		
Q	(states)	: set
Σ	(symbols)	: set
δ	(trans. rel.)	: $Q \times \Sigma \rightarrow \mathcal{P}(Q)$
q_0	(init. state)	: Q
F	(fin. states)	: $\mathcal{P}(Q)$
A transition is a triple $\langle q, c, q' \rangle$ with $q' \in \delta(q, c)$.		

Recontextualization: Translation Along Theory Morphisms



Recontextualization: Translation Along Theory Morphisms



Recontextualization: Implementation Approaches

String-Based Translation:

A **path** is a finite sequence e_1, \dots, e_n of edges with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.



A **path** is a finite sequence e_1, \dots, e_n of elements of $\{t \mid t \text{ is a transition}\}$ with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

Recontextualization: Implementation Approaches

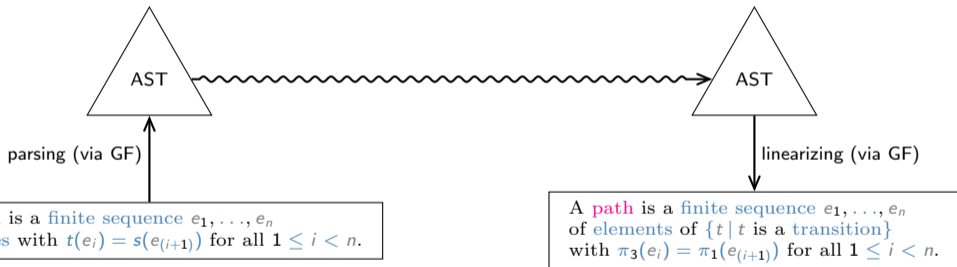
String-Based Translation:

A **path** is a **finite sequence** e_1, \dots, e_n of **edges** with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.



A **path** is a **finite sequence** e_1, \dots, e_n of **elements** of $\{t \mid t \text{ is a transition}\}$ with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

AST-Based Translation:



Recontextualization: Implementation Approaches

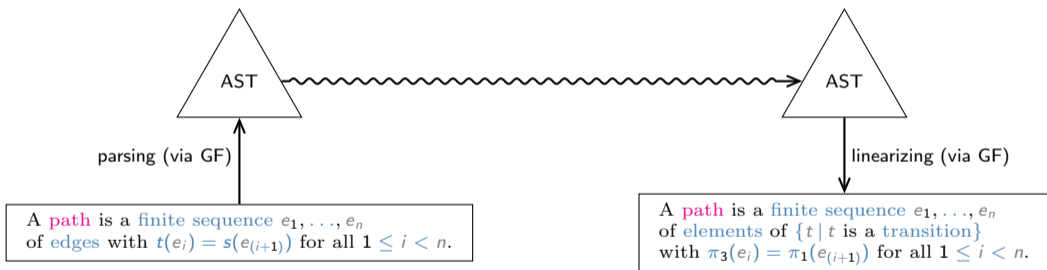
String-Based Translation:

A **path** is a finite sequence e_1, \dots, e_n of edges with $t(e_i) = s(e_{i+1})$ for all $1 \leq i < n$.



A **path** is a finite sequence e_1, \dots, e_n of elements of $\{t \mid t \text{ is a transition}\}$ with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

AST-Based Translation:



- ⇒ Grammatical correctness
- ⇒ Allows rewriting
- ⇒ Better scalability

Rewriting

A **path** is a finite sequence e_1, \dots, e_n of elements of $\{t \mid t \text{ is a transition}\}$ with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

Rewriting: Applying Natural Language Transformations

A **path** is a finite sequence e_1, \dots, e_n of elements of $\{t \mid t \text{ is a transition}\}$
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence t_1, \dots, t_n of transitions $t_k := \langle q_k, c_k, q'_k \rangle$
with $q'_i = q_{i+1}$ for all $1 \leq i < n$.

Rewriting: Applying Natural Language Transformations

A **path** is a finite sequence e_1, \dots, e_n of **transitions**
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence e_1, \dots, e_n of **elements of $\{t \mid t \text{ is a transition}\}$**
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence t_1, \dots, t_n of **transitions** $t_k := \langle q_k, c_k, q'_k \rangle$
with $q'_i = q_{i+1}$ for all $1 \leq i < n$.

Rewriting: Applying Natural Language Transformations

A **path** is a finite sequence e_1, \dots, e_n of transitions $e_k := \langle q_k, c_k, q'_k \rangle$
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence e_1, \dots, e_n of transitions
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence e_1, \dots, e_n of elements of $\{t \mid t \text{ is a transition}\}$
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence t_1, \dots, t_n of transitions $t_k := \langle q_k, c_k, q'_k \rangle$
with $q'_i = q_{i+1}$ for all $1 \leq i < n$.

Rewriting: Applying Natural Language Transformations

A **path** is a finite sequence e_1, \dots, e_n of transitions $e_k := \langle q_k, c_k, q'_k \rangle$
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

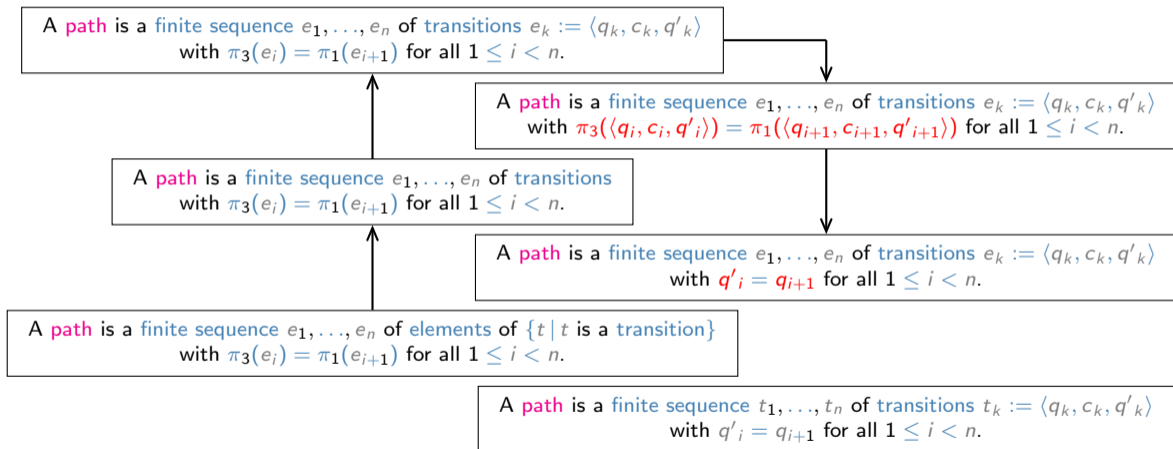
A **path** is a finite sequence e_1, \dots, e_n of transitions $e_k := \langle q_k, c_k, q'_k \rangle$
with $\pi_3(\langle q_i, c_i, q'_i \rangle) = \pi_1(\langle q_{i+1}, c_{i+1}, q'_{i+1} \rangle)$ for all $1 \leq i < n$.

A **path** is a finite sequence e_1, \dots, e_n of transitions
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

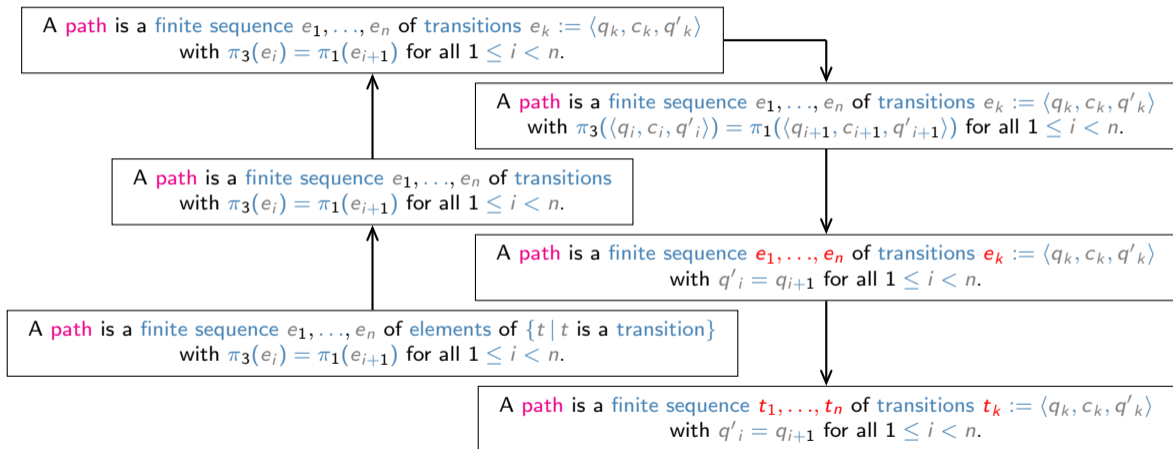
A **path** is a finite sequence e_1, \dots, e_n of elements of $\{t \mid t \text{ is a transition}\}$
with $\pi_3(e_i) = \pi_1(e_{i+1})$ for all $1 \leq i < n$.

A **path** is a finite sequence t_1, \dots, t_n of transitions $t_k := \langle q_k, c_k, q'_k \rangle$
with $q'_i = q_{i+1}$ for all $1 \leq i < n$.

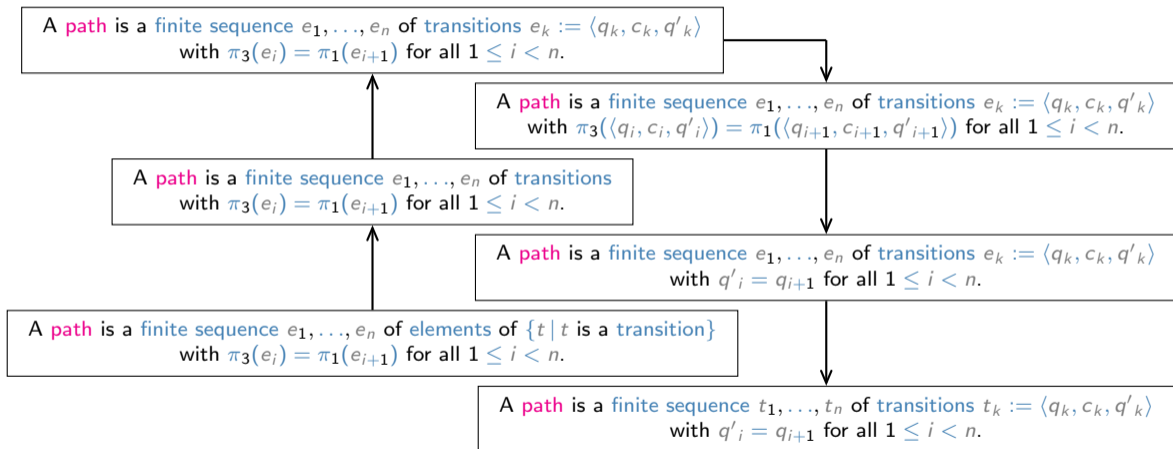
Rewriting: Applying Natural Language Transformations



Rewriting: Applying Natural Language Transformations



Rewriting: Applying Natural Language Transformations



Rewriting: Natural Language Transformation Rules

Archive of Transformation Rules: <https://flexiformal.github.io/rewriting-rules>

Rewriting: Natural Language Transformation Rules

Archive of Transformation Rules: <https://flexiformal.github.io/rewriting-rules>

“Comprehension Term Reduction”:

x : [⟨Var⟩]
y : ⟨Var⟩
z : ⟨CN⟩

“element x of {y | y is a z}”

“z x”

x = “”
y = “t”
z = “transition”

“element of {t | t is a transition}”

“transition”

Rewriting: Natural Language Transformation Rules

Archive of Transformation Rules: <https://flexiformal.github.io/rewriting-rules>

“Comprehension Term Reduction”:

x : $\langle \text{Var} \rangle$
y : $\langle \text{Var} \rangle$
z : $\langle \text{CN} \rangle$

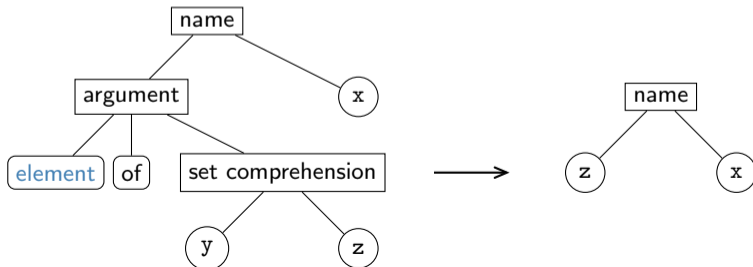
“element x of {y | y is a z}”

“z x”

x = “”
y = “t”
z = “transition”

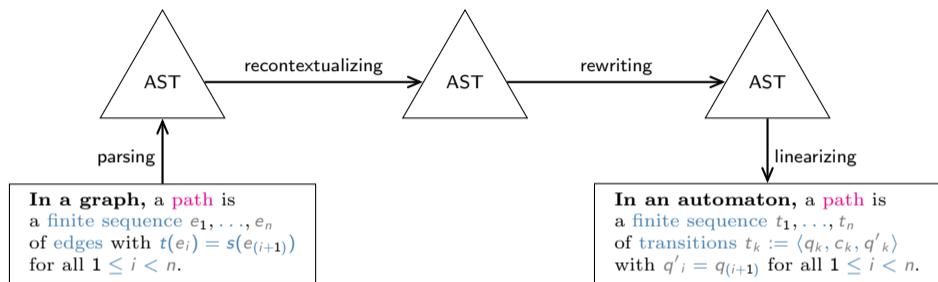
“element of {t | t is a transition}”

“transition”



Conclusion

Conclusion: An Improved Recontextualization Mechanism



- ▶ Semantic GF-based grammar
- ▶ Collection of rewriting rules
- ⇒ Better scalability
- ⇒ Grammatical correctness
- ⇒ Increased naturality

Archive of Rewriting Rules:

<https://flexiformal.github.io/rewriting-rules>

Proof of Concept Implementation:

<https://github.com/FlexiFormal/relocalization>