Partial Algebra & Semantic analysis

Adrien Ragot[1,2]

1: Université Sorbonne Paris-Nord

2: Università degli Studi Roma tre

I Overview

II Indexed Partial Magmas

III Conclusion

I Overview

II Indexed Partial Magmas

III Conclusion

I Overview

1- Genesis

2 - Syntax & semantics

3 - Correctness framework

4 - In compiler design

I Overview

1- Genesis

2 - Syntax & semantics

3 - Correctness framework

4 - In compiler design

GENESIS

Phd Project

Proof nets + Realisability

GENESIS

Phd Project

Proof nets + Realisability

correctness criteria MLL2
(parsing / coalescence)

GENESIS

Phd Project

Proof nets + Realisability

combinatorial, hard to properly prove

correctness criteria MLL2 (parsing / coalescence)

GENESIS

PhD Project

Proof nets + Realisability

combinatorial, hard to properly prove

correctness criteria MLL2 (parsing / coalescence)

extract algebraically structured arguments

GENESIS

Phd Project

Proof nets + Realisability

combinatorial, hard to properly prove

correctness criteria MLL2 (parsing / coalescence)

extract algebraically structured arguments

personal pref. esthetic

generalisation

more Robust

GENESIS

Phd Project

Proof nets + Realisability

combinatorial, hard to properly prove

correctness criteria MLL2
(parsing / coalescence)

personal pref.
esthetic

generalisation

more Robust

extract algebraically structured arguments

Indexed Partial Algebra

I Overview

1 - Genesis

2 - Syntax & semantics

3 - Correctness framework

4 - In compiler design

Syntax & semantics

Syntax & semantics

Syntax & semantics

Syntax & semantics

Syntactical domain

types = desc. of event A

terms = schemes to compose descr. t

semantical domain

Syntax & semantics

Syntactical domain

types = desc. of event A

terms = schemes to compose descr. t

semantical domain

realisat. of the event

Syntax & semantics

Syntactical domain

types = desc. of event A

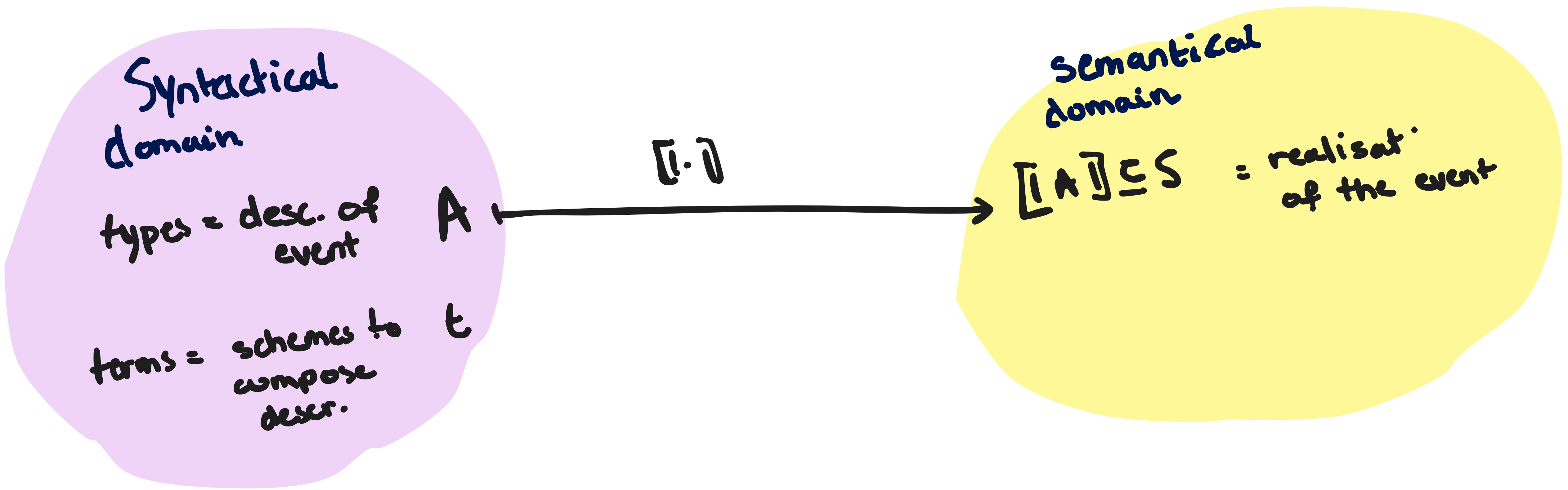
terms = schemes to compose descr. t

semantical domain

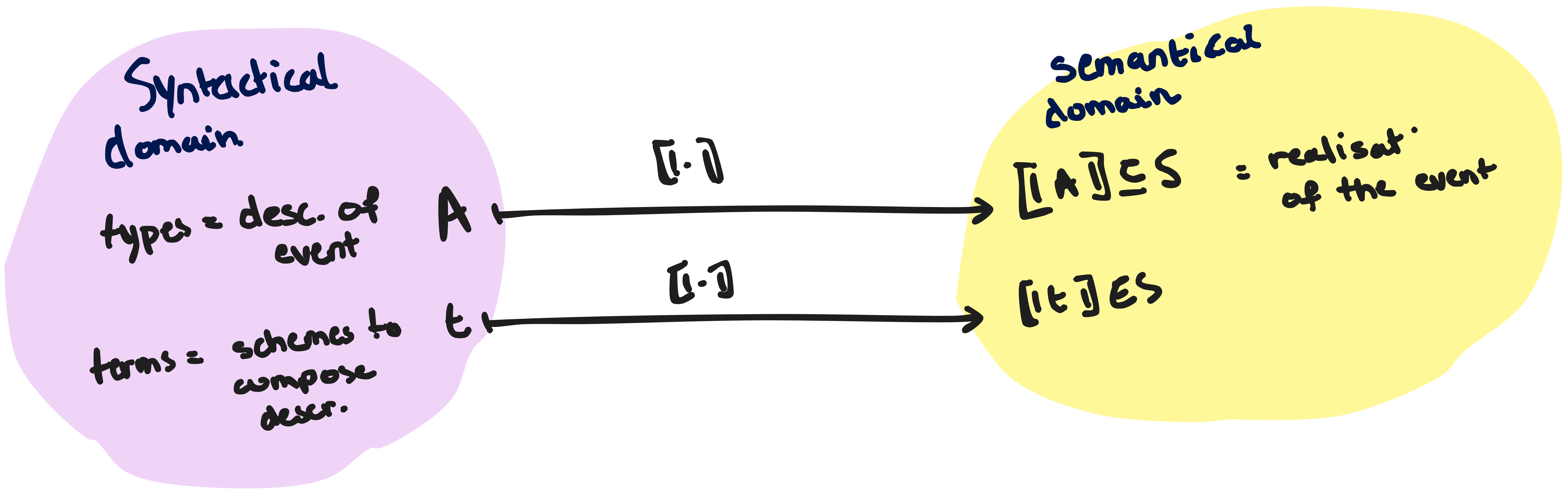
realisat. of the event

how do they relate?

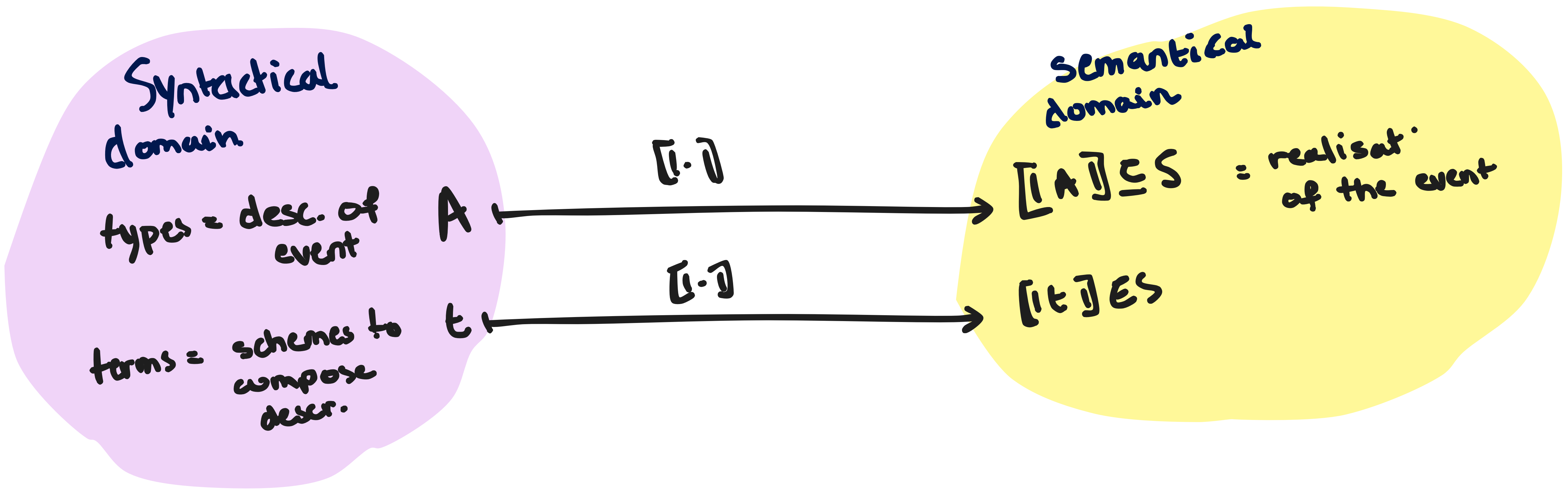
Syntax & semantics



Syntax & semantics

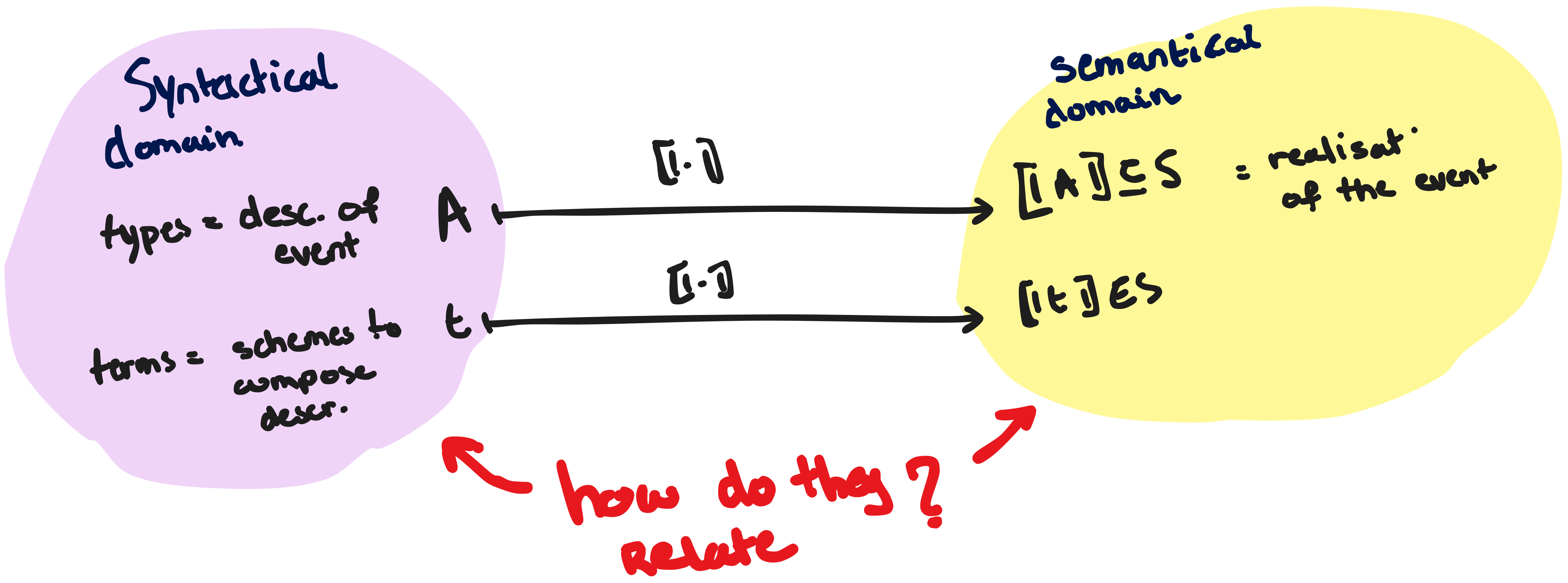


Syntax & semantics



soundness : $t \vdash A \Longrightarrow [\![t]\!] \in [\![A]\!]$

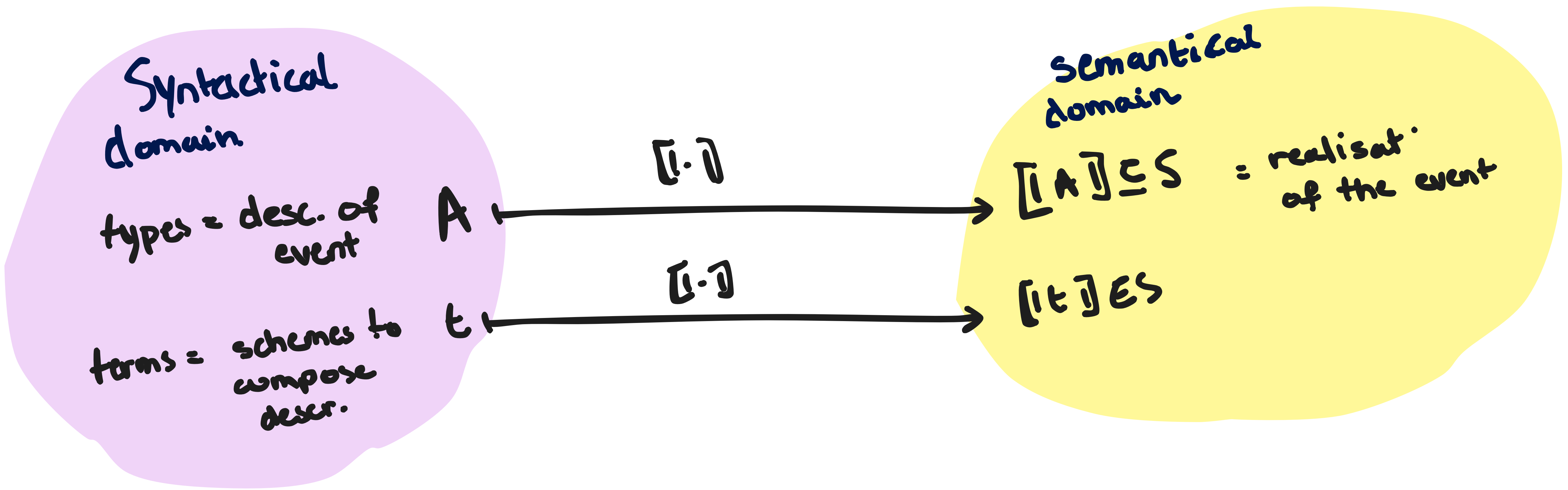
Syntax & semantics



soundness : $t \vdash A \implies [\![t]\!] \in [\![A]\!]$

completeness : $t \in [\![A]\!] \implies \exists u \in \mathrm{SYN},\ [\![u]\!] = t \wedge u \vdash A$

Syntax & semantics



In some cases: both SYN and SEM are syntactical

Syntax & semantics

Syntactical domain

types = desc. of event

terms = schemes to compose descr.

$A \xrightarrow{[\![\cdot]\!]} [\![A]\!] \subseteq S$ = realisat. of the event

$t \xrightarrow{[\![\cdot]\!]} [\![t]\!] \in S$

Semantical domain

STλ derivation tree, choreo, proof tree

λ-calculus, pi-calculus, KAM config, ...

In some cases: both SYN and SEM are syntactical

I Overview

1- Genesis

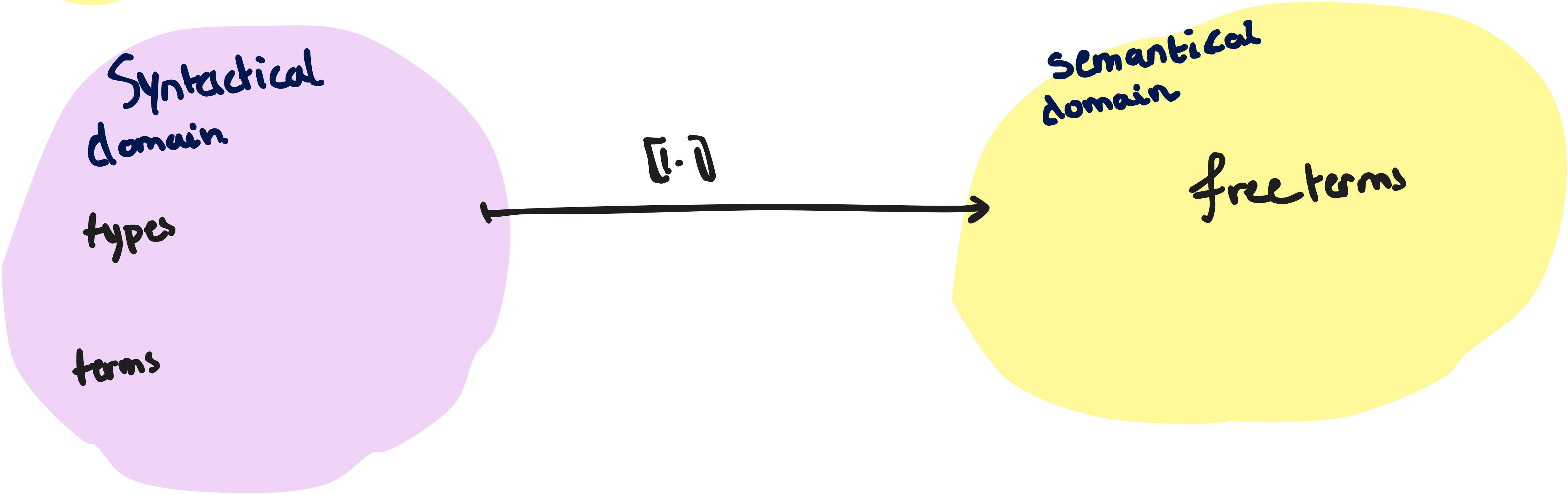
2 - Syntax & semantics

3 - Correctness framework

4 - In compiler design

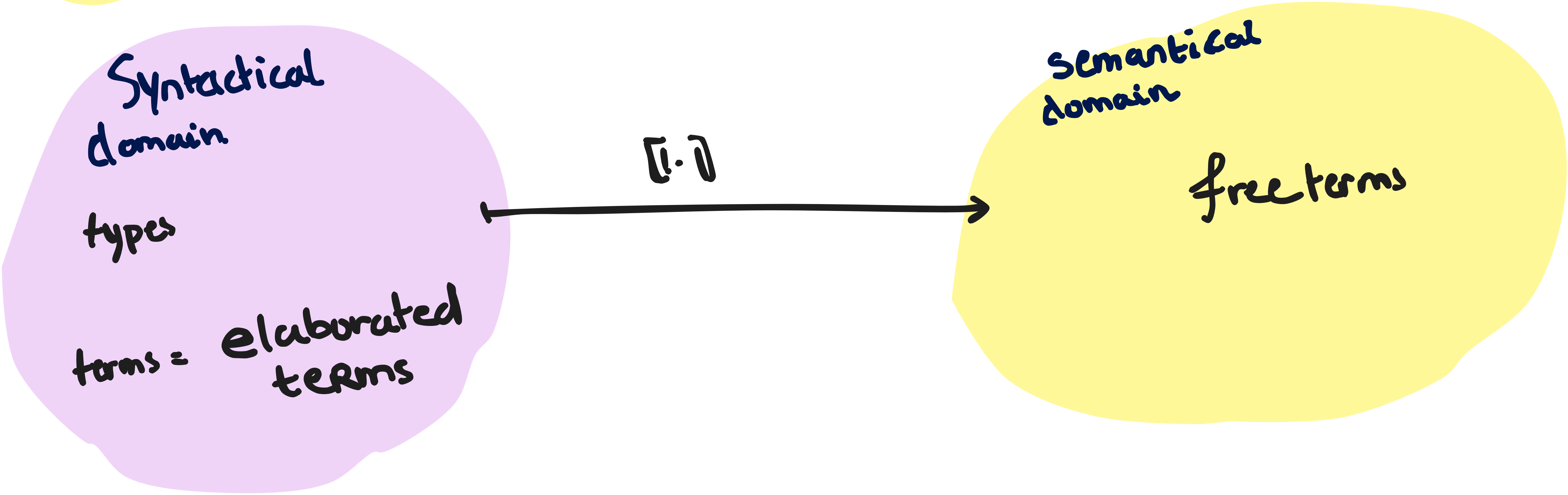
The Correctness Framework

both SYN and SEM are syntactical



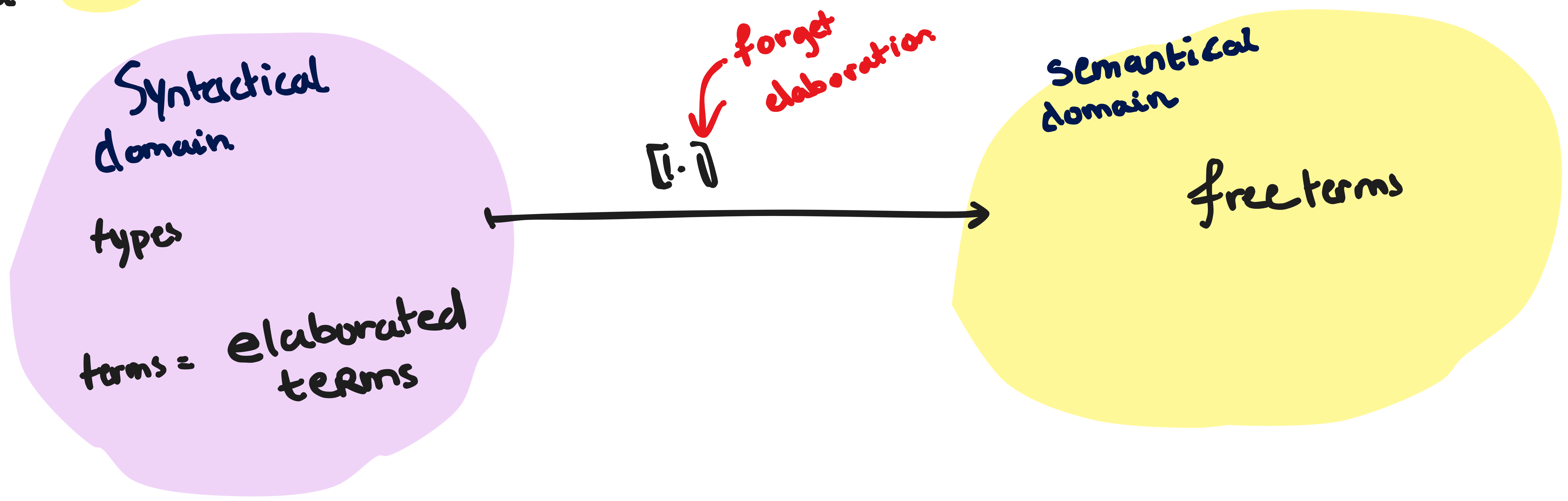
The Correctness Framework

both SYN and SEM are syntactical



The Correctness Framework

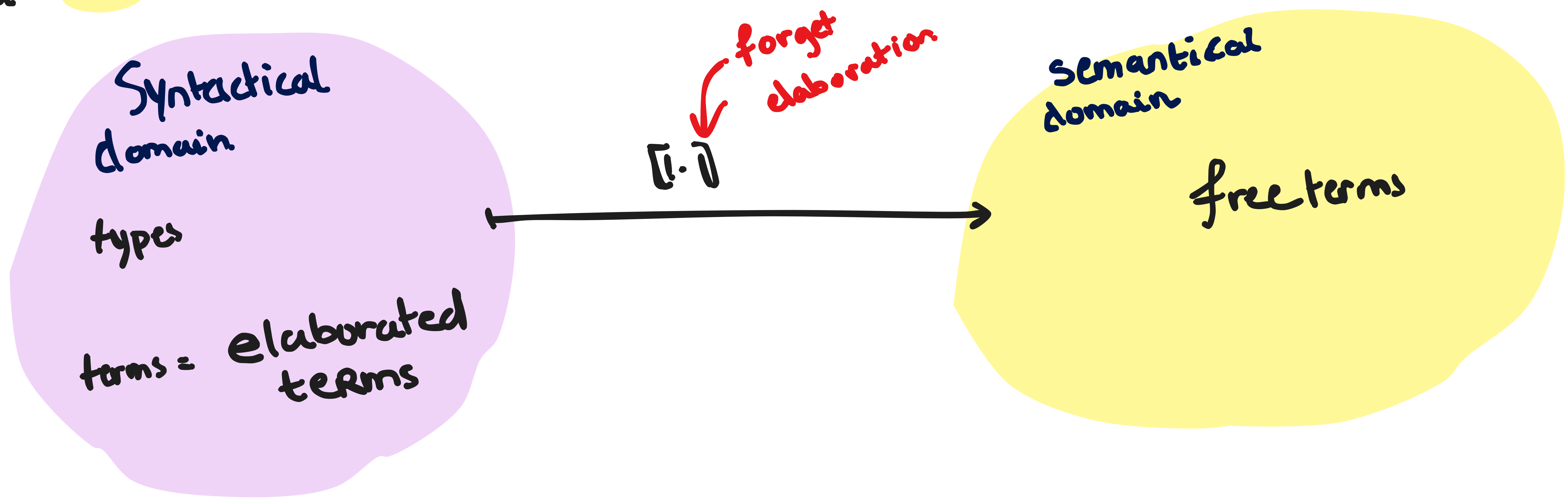
both SYN and SEM are syntactical



The Correctness Framework

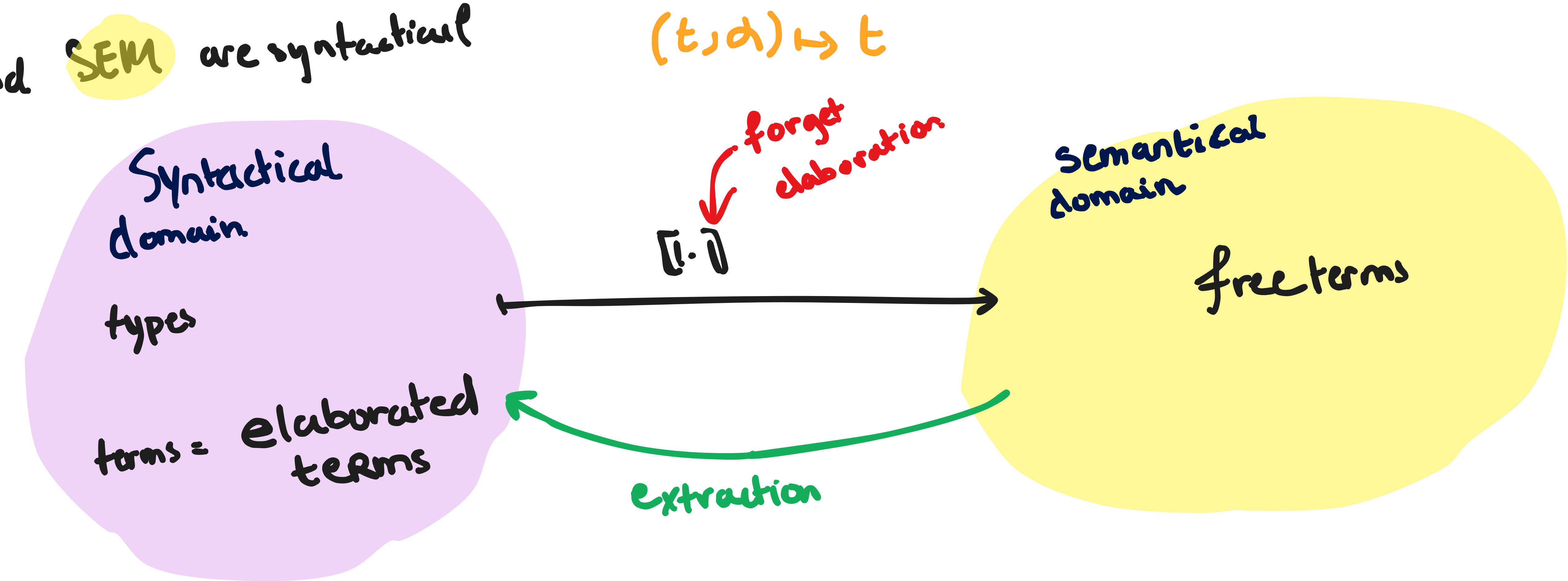
both SYN and SEM are syntactical

$(t, d) \mapsto t$



The Correctness Framework

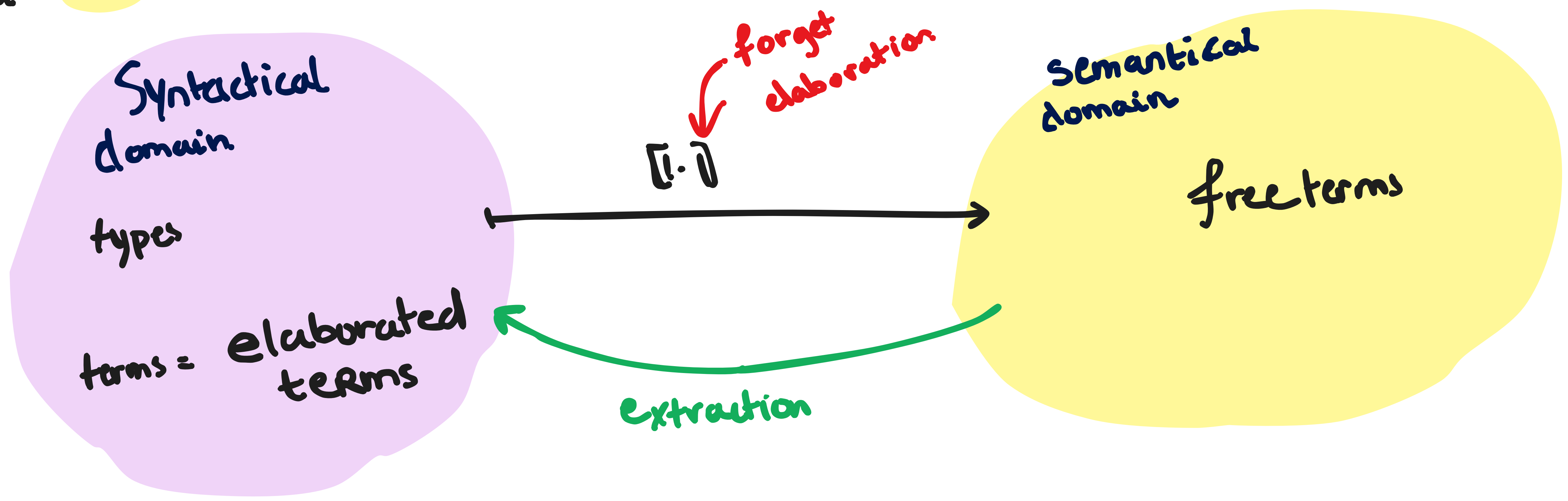
both SYN and SEM are syntactical



The Correctness Framework

both SYN and SEM are syntactical

$(t, d) \mapsto t$

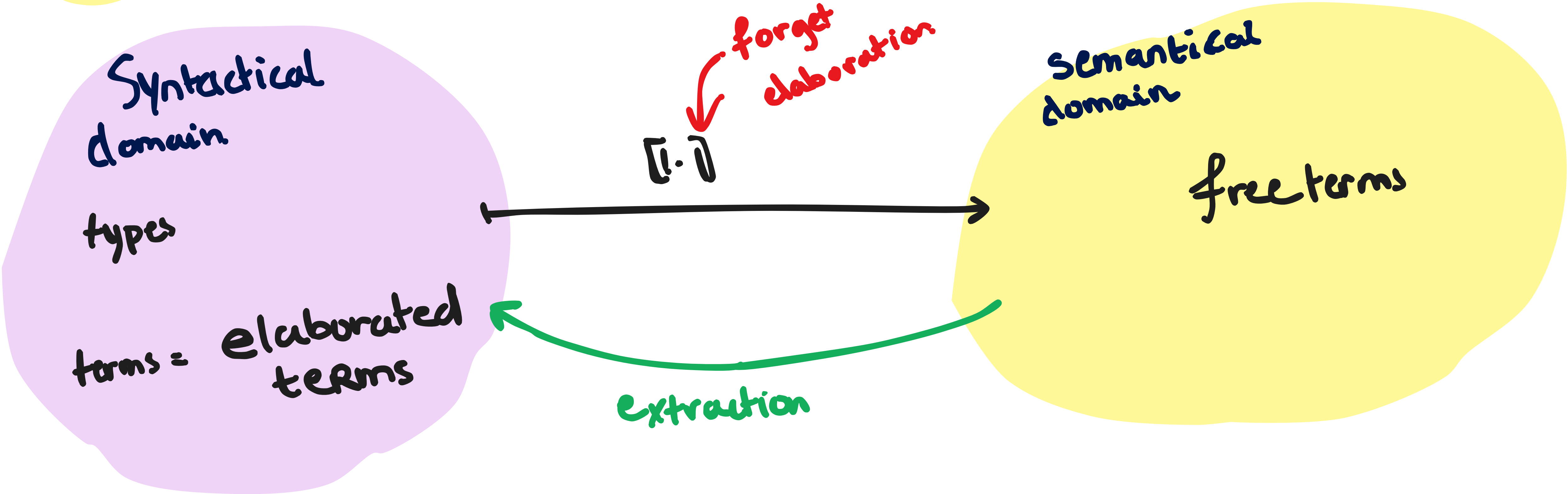


$$[\![A]\!] = \{ t \in \text{SEM} \mid \vdash \rightarrow t^* \vdash A \}$$

The Correctness Framework

both SYN and SEM are syntactical

$(t, \alpha) \mapsto t$



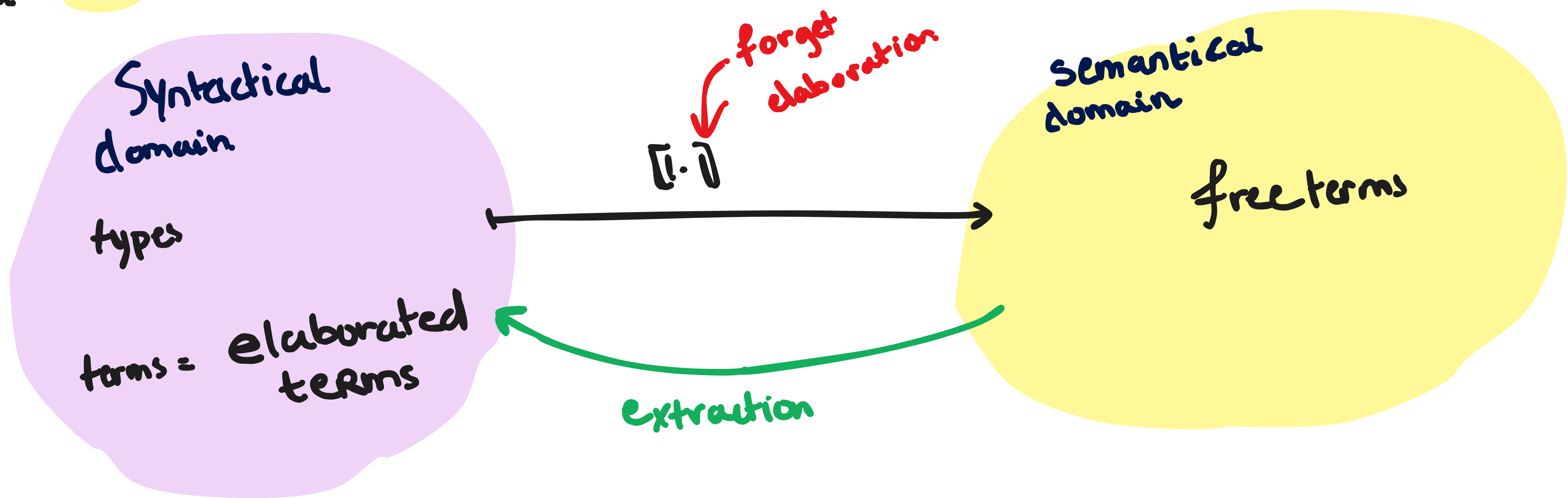
$$[\![A]\!] = \{ t \in \text{SEM} \mid t \rightarrow t^* \vdash A \}$$

$$[\![t^* \vdash A]\!] = t \quad , \quad t^* = (t, \alpha)$$

The Correctness Framework

both SYN and SEM are syntactical

$(t, \alpha) \mapsto t$



$$[\![A]\!] = \{ t \in \mathrm{SEM} \mid t \rightarrow t^* \vdash A \}$$

$$[\![t^* \vdash A]\!] = t \quad , \quad t^* = (t, \alpha)$$

example : SYN = STλ $[\![\cdot]\!]$ = take proof-term

SEM = λ-calc

$\rightarrow$ = type inference

I Overview

1 - Genesis

2 - Syntax & semantics

3 - Correctness framework

4 - In compiler design

In Compiler design

compiler = translator

In Compiler design

compiler = translator

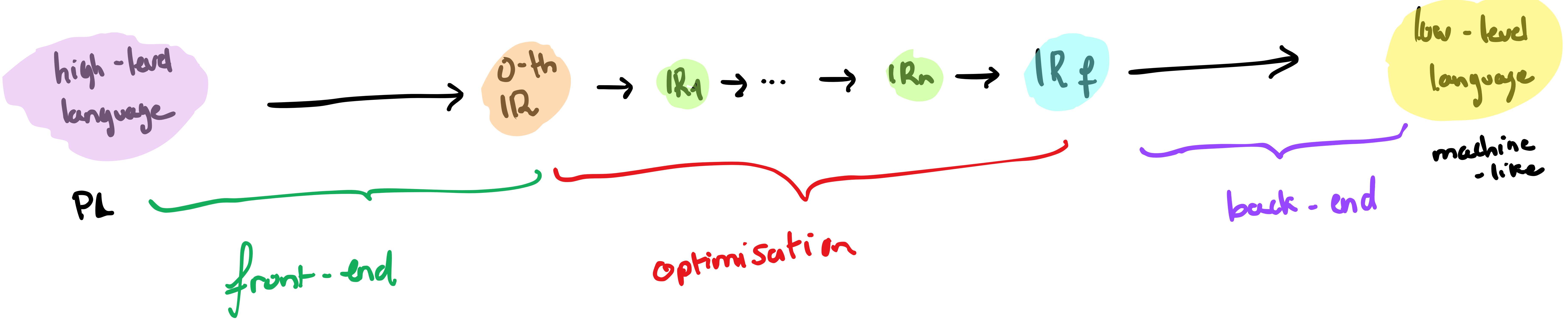
high-level language $\longrightarrow$ low-level language

PL

machine-like

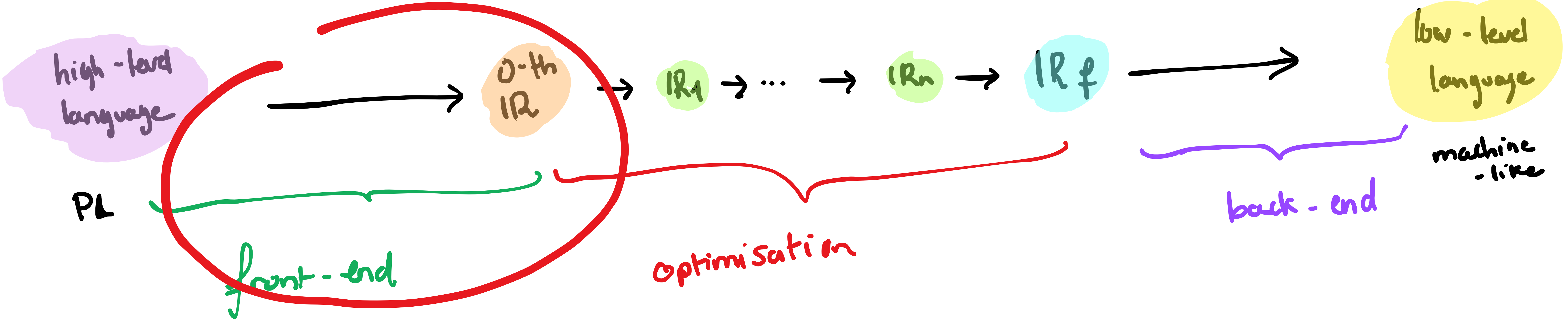
In Compiler design

compiler = translator

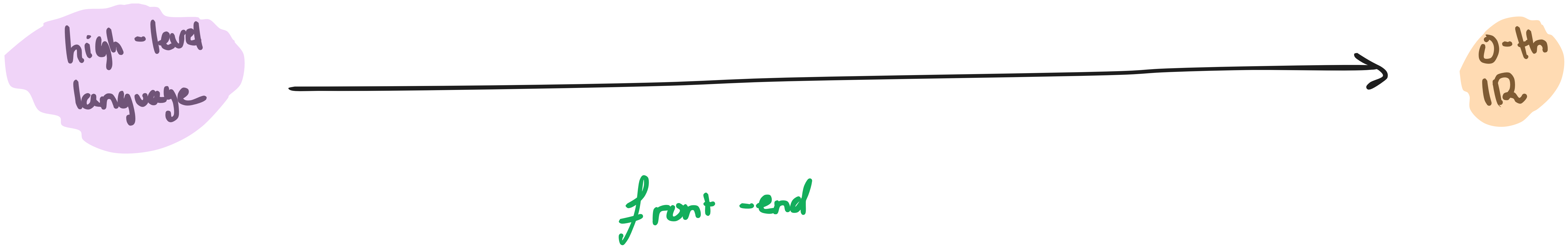


In Compiler design

compiler = translator



In Compiler design



In Compiler design

high-level language $\xrightarrow{\text{front-end}}$ 0-th IR

= Stream of char

= AST / term + elaboration

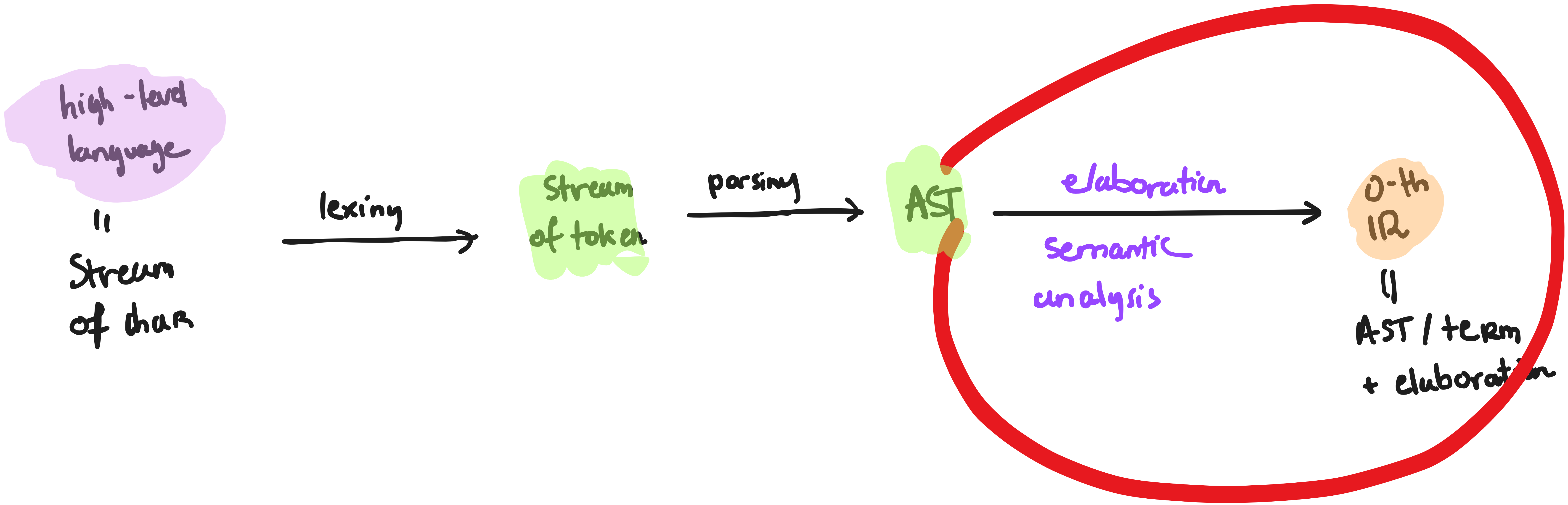
In Compiler design

high-level language
=
Stream of char

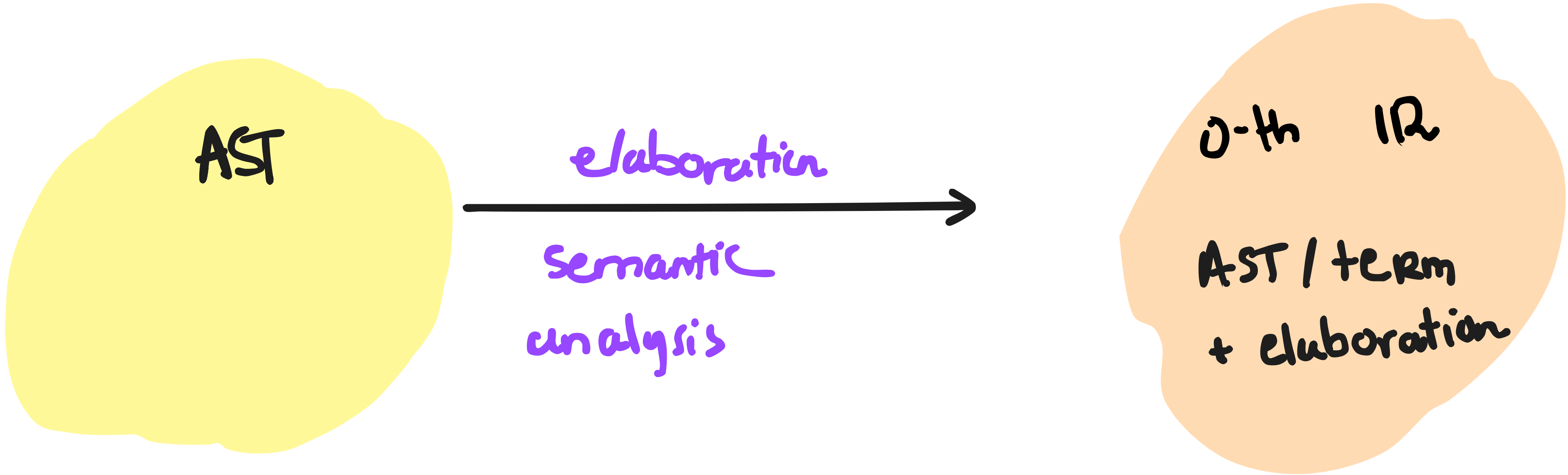
$\xrightarrow{\text{lexing}}$ Stream of token $\xrightarrow{\text{parsing}}$ AST $\xrightarrow[\text{semantic analysis}]{\text{elaboration}}$ 0-th IR

0-th IR
=
AST / term + elaboration

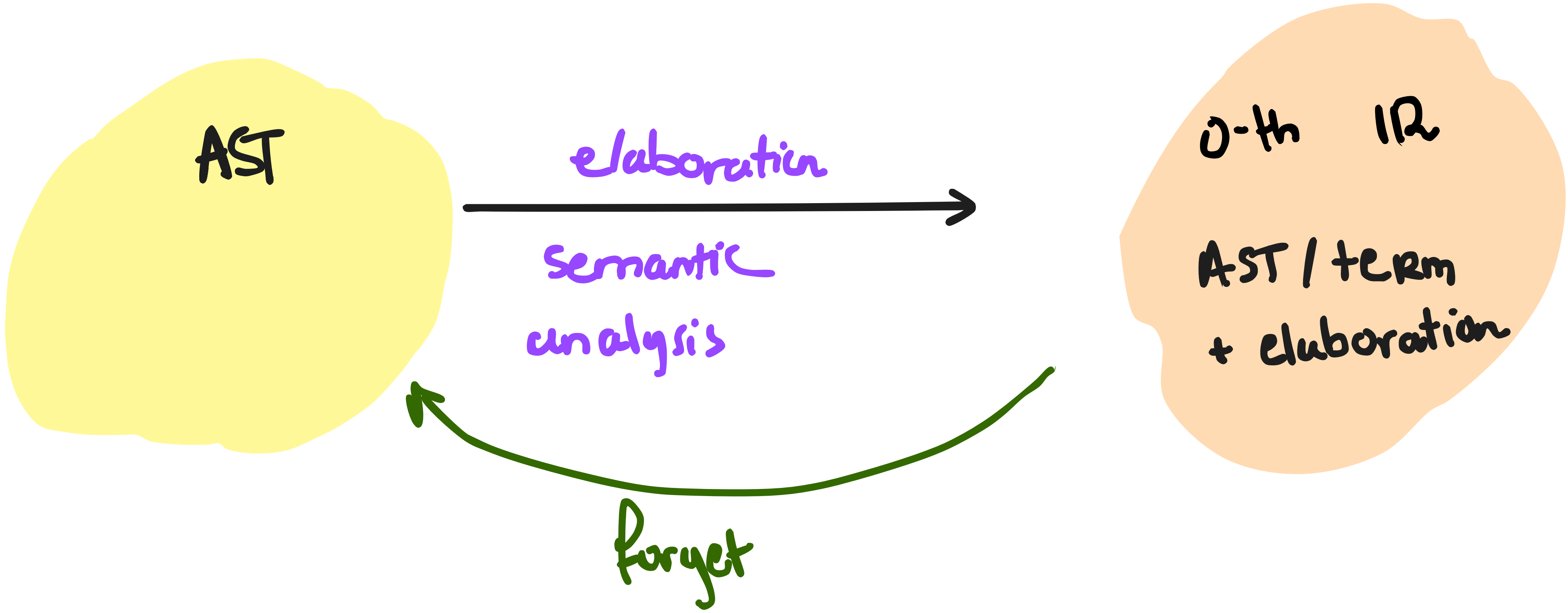
In Compiler design



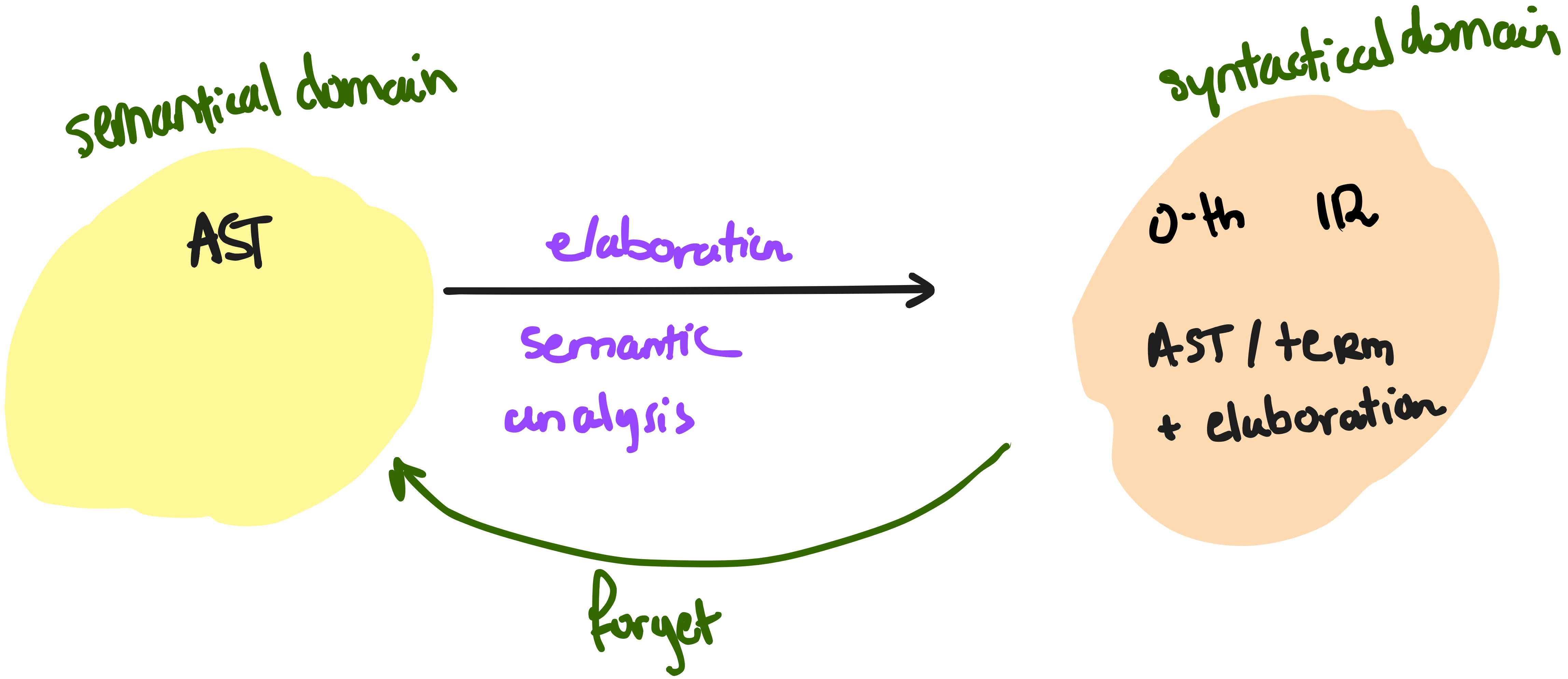
In Compiler design



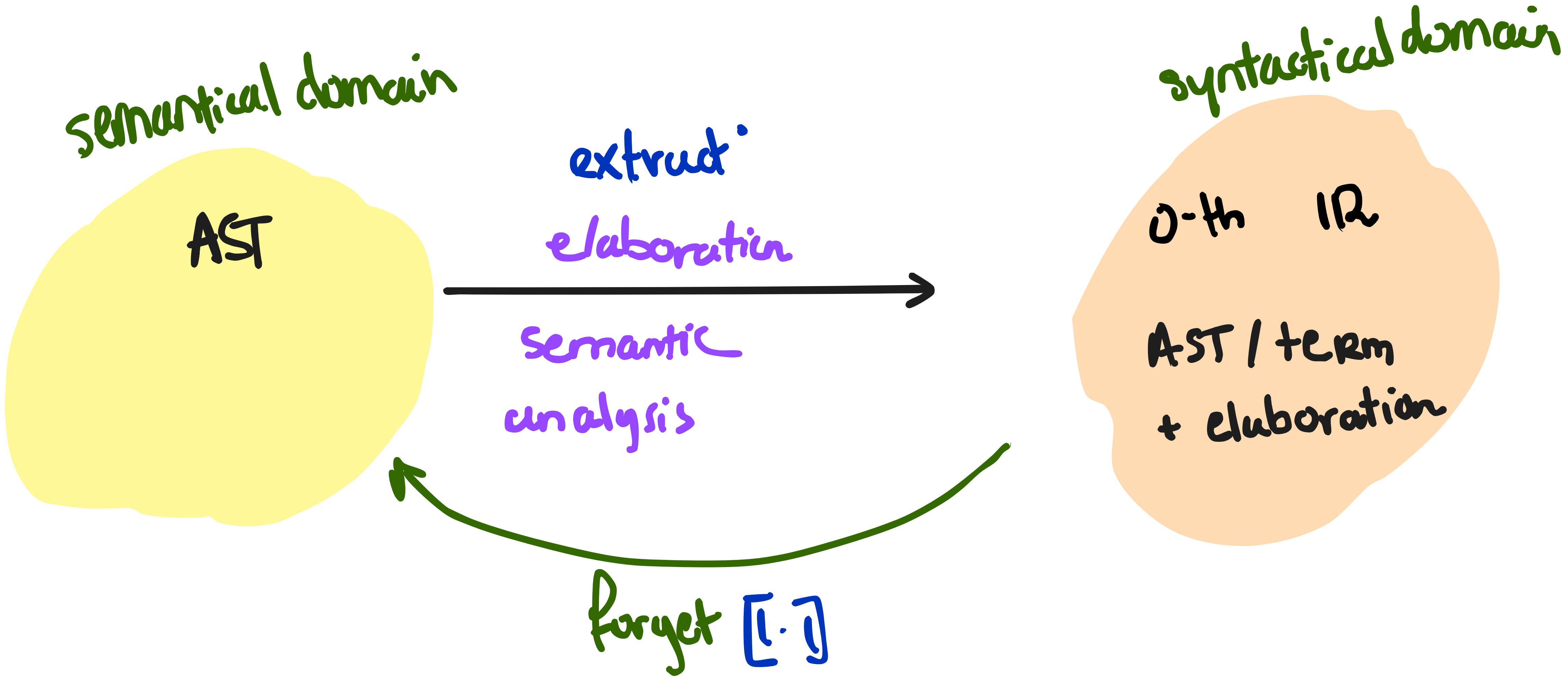
In Compiler design



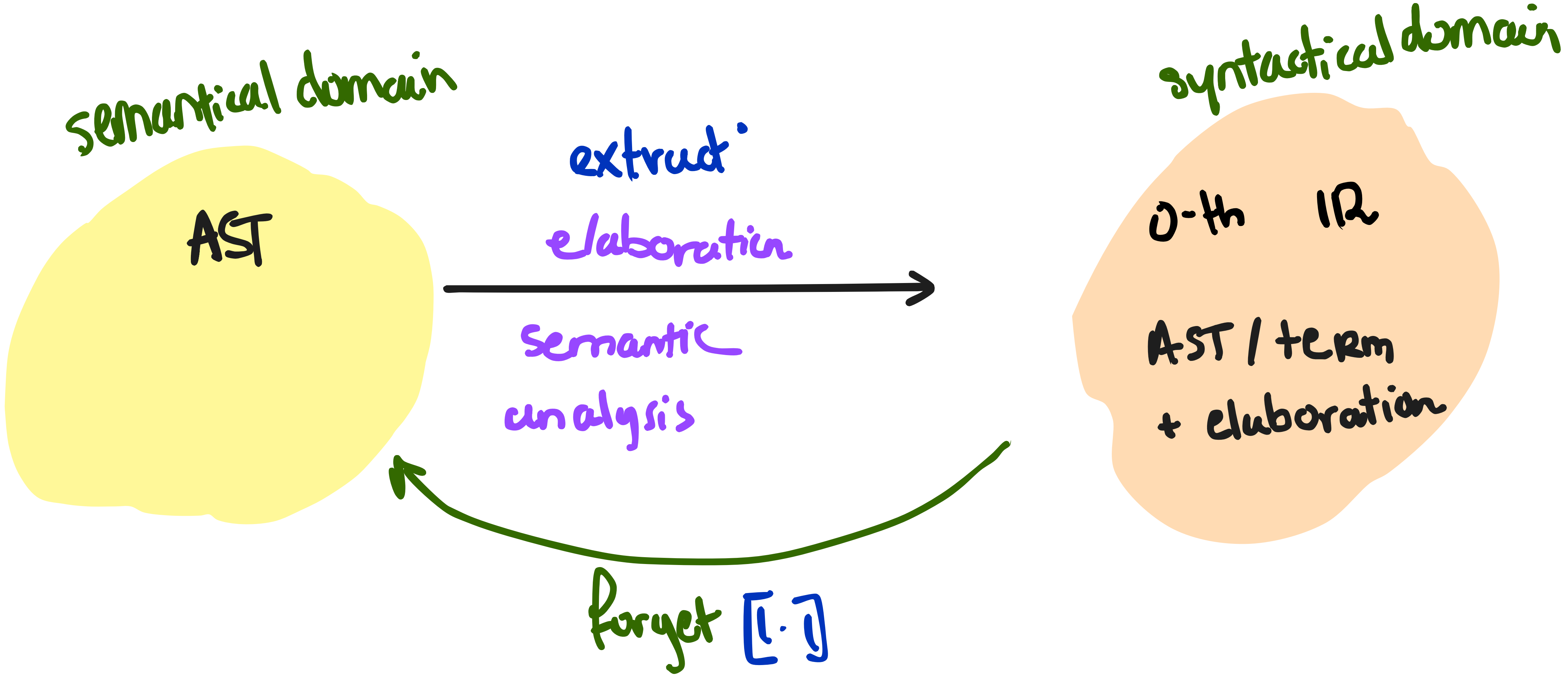
In Compiler design



In Compiler design



In Compiler design



correctness framework

I Overview

II Indexed Partial Magmas

III Conclusion

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

↳ running example

3 - Morphisms

↳ running example

4 - confluence of coalescence

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

↳ running example

3 - Morphisms

↳ running example

4 - confluence of coalescence

MAIN RESULTS

THM 1

$x \in gen_A(B) \iff \exists a \in A, \; x \to_B a$

MAIN RESULTS

THM 1

$$x \in \mathrm{gen}_A(B) \iff \exists a \in A, \quad x \to_B a$$

?

?

MAIN RESULTS

THM 1

$x \in \text{gen}_A(B) \iff \exists a \in A,\ x \xrightarrow{}_B a$

? ?

THM 2

if f is sound and complete (w.r.t. R)

and R maps $\text{gen}_A B$ to $\text{gen}_A B$

$f(\text{gen}_A(B)) = \text{gen}_{f(A)}(f(B))$

MAIN RESULTS

THM 1

?

$$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \to_B a$$

?

THM 2

? ?

if f is sound and complete (w.r.t. R) and R maps $\mathrm{gen}_A B$ to $\mathrm{gen}_A B$

$$f(\mathrm{gen}_A(B)) = \mathrm{gen}_{f(A)}(f(B))$$

MAIN RESULTS

THM 1

$x \in \underline{gen_A(B)} \iff \exists a \in A, \; x \to_B a$

?

?

THM 2

? ?

if f is sound and complete (w.r.t. R) and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

MAIN RESULTS

THM 1

$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \to_B a$

?

?

THM 2

?

?

if f is sound and complete (w.r.t. R)
and R maps $\mathrm{gen}_A B$ to $\mathrm{gen}_A B$

$f(\mathrm{gen}_A(B)) = \mathrm{gen}_{f(A)}(f(B))$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

?

?

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

↳ running example

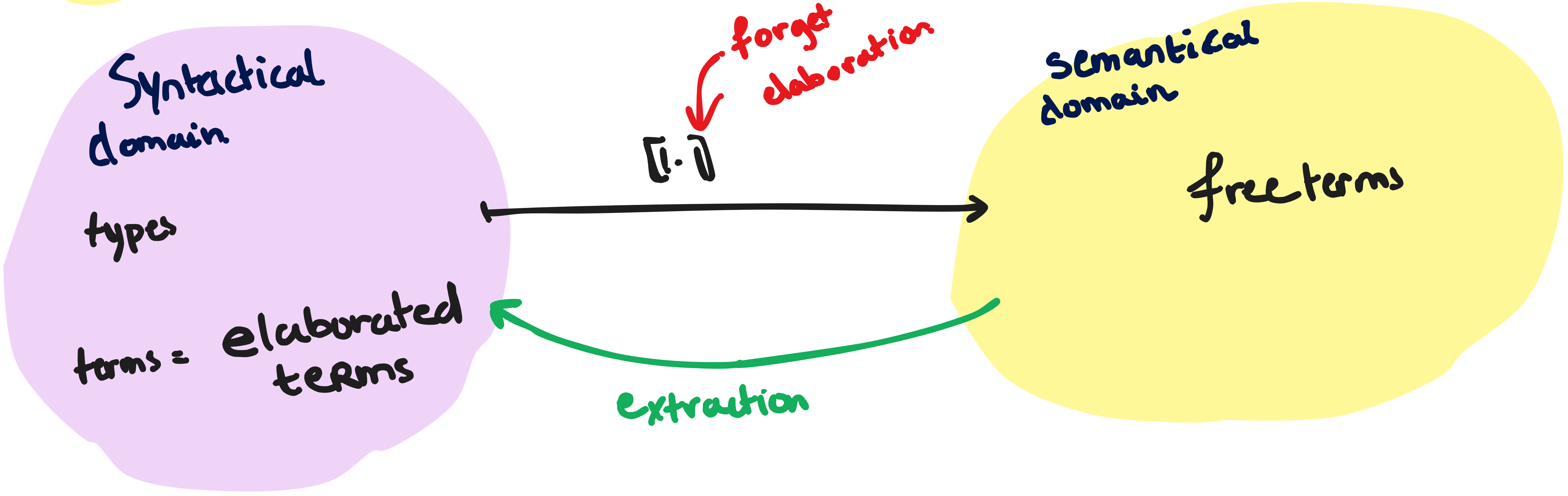
3 - Morphisms

↳ running example

4 - confluence of coalescence

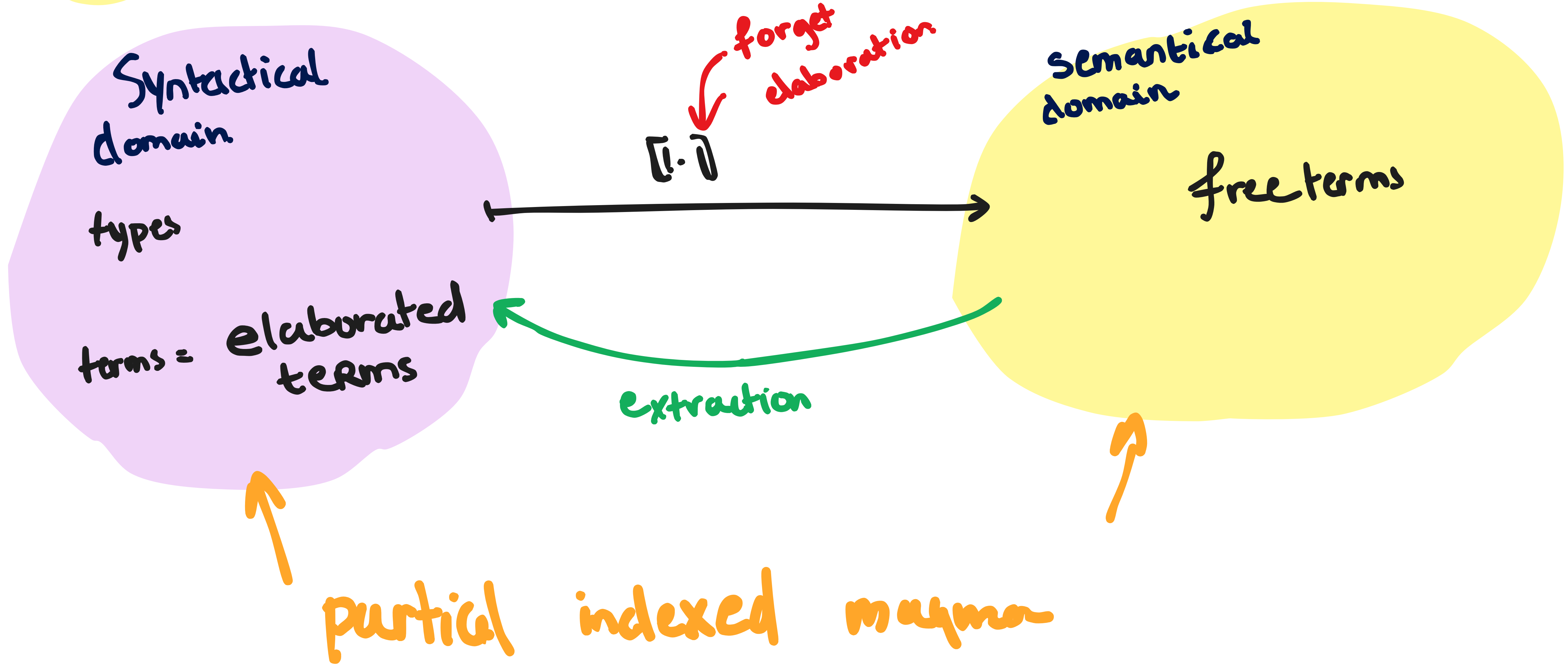
Partial indexed Magma

both SYN and SEM are syntactical

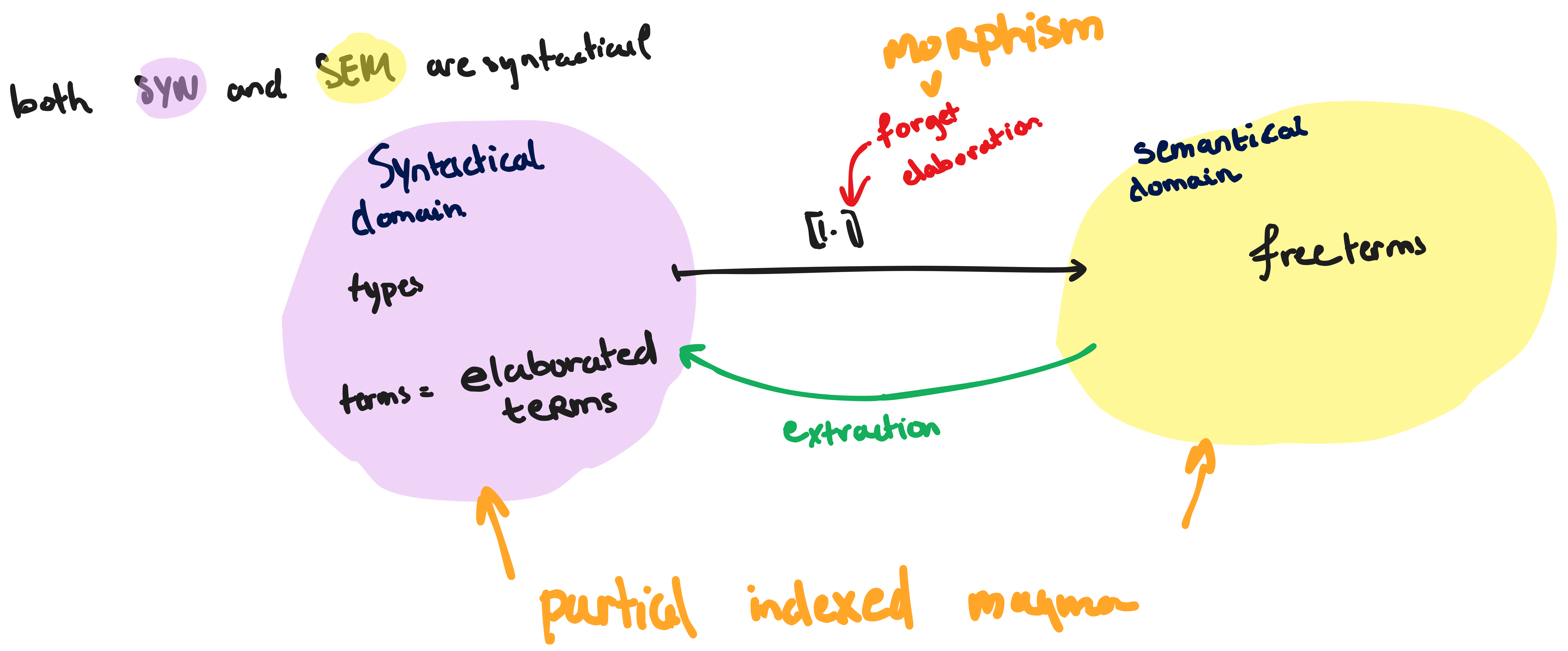


Partial indexed Magma

both SYN and SEM are syntactical



Partial indexed Magma



Partial indexed Magma

Partial indexed Magma

$$(X, \ll, I)$$

Partial indexed Magma

$$(X, \ll, I)$$

set

Partial indexed Magma

$$(X, \ll, I)$$

Set

$\ll : I \times X \times X \to X$

Partial

Partial indexed Magma

$$(X, \ll, I)$$

set

set

$\ll : I \times X \times X \to X$

<u>partial</u>

Partial indexed Magma

$$(\underset{\text{set}}{X}, \underset{\text{partial}}{\ll}, \underset{\text{set}}{I})$$

$\ll : I \times X \times X \to X$

Partial

- $x \ll_i y$ not always defd.
-

Partial indexed Magma

$$(X, \ll, I)$$

Set

set

$\ll : I \times X \times X \to X$

Partial

- $x \ll_i y$ not always defd.
- $\ll_i$ models substitution at i
-

Partial indexed Magma

$$(X, \ll, I)$$

set (for X), set (for I)

$\ll : I \times X \times X \to X$

<u>Partial</u>

- $x \ll_i y$ not always defd.
- $\ll_i$ models substitution at i
- typing derivation π as

  $\pi_1 \ll_{e_1} \pi_2 \ll_{e_2} \cdots \ll_{e_n} \pi_{n+1}$

Partial indexed Magma

$$(X, \ll, I)$$

set

set

set

$\ll : I \times X \times X \to X$

<u>partial</u>

$gen_A(B) =$

Partial indexed Magma

$$(X, \ll, I)$$

set   set

$\ll : I \times X \times X \to X$

Partial

$gen_A(B) =$

smallest $U \subseteq X$ such that

1- $A \subseteq U$

2- $\forall i \in I,\ \forall b \in B,\ \forall u \in U$

$u \ll_i b \downarrow \implies u \ll_i b \in U$

Partial indexed Magma

$\to_B$ = $x \to_B x'$

iff $\exists i \in I, \exists b \in B;$

$x = x' \ll_i b$

$gen_A(B)$ =

smallest $U \subseteq X$ such that

1- $A \subseteq U$

2- $\forall i \in I, \forall b \in B, \forall u \in U$

$u \ll_i b \downarrow \Rightarrow u \ll_i b \in U$

Partial indexed Magma

$\to_B$ = $x \to_B x'$

iff $\exists i \in I, \exists b \in B;$

$x = x' \ll_i b$

$gen_A(B)$ =

smallest $U \subseteq X$ such that

1- $A \subseteq U$

2- $\forall i \in I, \forall b \in B, \forall u \in U$

$u \ll_i b \downarrow \Rightarrow u \ll_i b \in U$

THM 1

$x \in gen_A(B) \iff \exists a \in A, \; x \to_B a$

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

↳ running example

3 - Morphisms

↳ running example

4 - confluence of coalescence

RUNNING EXAMPLE

RUNNING EXAMPLE

$(\lambda_c, <<, \mathrm{VAR})$ is a partial ind. magma

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial ?

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial ?

$\ll$ preserve well-scope

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial ?

$\ll$ preserve well-scope

$$(\lambda x . x)\, y \ll_x x = \bot$$

$$(\lambda x . x)\, y \ll_x 3 = (\lambda x . x)\, 3$$

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial ?

$\ll$ preserve well-scope

$$(\lambda x.x)y \ll_x x = \bot$$

$$(\lambda x.x)y \ll_x 3 = (\lambda x.x)3$$

$\ll$ preserve SN

Running Example

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial ?

$\ll$ preserve well-scope

$$(\lambda x.x)y \ll_x x = \bot$$

$$(\lambda x.x)y \ll_x 3 = (\lambda x.x)3$$

$\ll$ preserve SN

$$x \ll_x \lambda x(x)x = \lambda x.(x)x$$

$$(\lambda x.(x)x)\, y \ll_y \lambda z(z)z = \bot$$

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial ?

$\ll$ preserve wel-scope

$$(\lambda x . x) y \ll_x x = \bot$$

$$(\lambda x . x) y \ll_x 3 = (\lambda x . x) 3$$

$\ll$ preserve SN

$$x \ll_x \lambda x (x) x = \lambda x . (x) x$$

$$(\lambda x . (x) x) y \ll_y \lambda z (z) z = \bot$$

$\ll$ preserve STλ

Running Example

$(\lambda_c , \ll , \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

why partial?

$\ll$ preserve wel-scope

$$(\lambda x.x)y \ll_x x = \bot$$

$$(\lambda x.x)y \ll_x 3 = (\lambda x.x)3$$

$\ll$ preserve SN

$$x \ll_x \lambda x (x)x = \lambda x.(x)x$$

$$(\lambda x.(x)x)\, y \ll_y \lambda z.(z)z = \bot$$

$\ll$ preserve STλ

$$(x)y \ll_y x = \bot$$

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v, \lambda x . v, (v)u \mid v, u \in \mathrm{VAR} \}$$

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v, \lambda x . v, (v)u \mid v, u \in \mathrm{VAR} \}$$

$$\lambda_c = \mathrm{gen}_{\mathrm{VAR}}(\lambda_0)$$

RUNNING EXAMPLE

$(\lambda_c, \ll, VAR)$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v, \lambda x . v, (v) u \mid v, u \in VAR \}$$

$$\lambda_c = gen_{VAR}(\lambda_0)$$

$\lambda x . x + 3$

Running Example

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v,\ \lambda x . v,\ (v)u \mid v, u \in \mathrm{VAR} \}$$

$$\lambda_c = \mathrm{gen}_{\mathrm{VAR}}(\lambda_0)$$

$$\lambda x . x + 3 = e_0 \ll_{e_0} \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

RUNNING EXAMPLE

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v, \lambda x . v, (v)u \mid v, u \in \mathrm{VAR} \}$$

$$\lambda_c = \mathrm{gen}_{\mathrm{VAR}}(\lambda_0)$$

$$\lambda x . x+3 = e_0 \ll_{e_0} \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

Running Example

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v, \lambda x . v, (v)u \mid v, u \in \mathrm{VAR} \}$$

$$\lambda_c = \mathrm{gen}_{\mathrm{VAR}}(\lambda_0)$$

$$\lambda x . x+3 = e_0 \ll_{e_0} \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_2 + e_3 \ll_{e_3} x \ll_{e} 3$$

RUNNING EXAMPLE

$(\lambda_c, \ll, VAR)$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v, \lambda x . v, (v)u \mid v, u \in VAR \}$$

$$\lambda_c = \mathrm{gen}_{VAR}(\lambda_0)$$

$$\lambda x . x+3 = e_0 \ll_{e_0} \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_2 + e_3 \ll_{e_2} x \ll_{e} 3$$

$$= \lambda x . x + e_3 \ll_{e_3} 3$$

Running Example

$(\lambda_c, \ll, \mathrm{VAR})$ is a partial ind. magma

$$t \ll_x u = t[x \mapsto u]$$

$$\lambda_0 = \{ v,\ \lambda x . v,\ (v)u \mid v, u \in \mathrm{VAR} \}$$

$$\lambda_c = \mathrm{gen}_{\mathrm{VAR}}(\lambda_0)$$

$$\lambda x . x+3 = e_0 \ll_{e_0} \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_1 \ll_{e_1} e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . e_2 + e_3 \ll_{e_2} x \ll_{e_3} 3$$

$$= \lambda x . x + e_3 \ll_{e_3} 3$$

$$= \lambda x . x + 3$$

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

↳ running example

3 - Morphisms

↳ running example

4 - confluence of coalescence

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

morphism : $f(x \ll_i y)\downarrow$ and $f(x) <_i f(y)\downarrow \implies f(x \ll_i y) = f(x) <_i f(y)$

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

morphism : $f(x \ll_i y)\downarrow$ and $f(x) <_i f(y)\downarrow \implies f(x \ll_i y) = f(x) <_i f(y)$

sound: $f(x \ll_i y)\downarrow \implies f(x) <_i f(y)\downarrow$

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

morphism : $f(x \ll_i y)\downarrow$ and $f(x) <_i f(y)\downarrow \implies f(x \ll_i y) = f(x) <_i f(y)$

sound: $f(x \ll_i y)\downarrow \implies f(x) <_i f(y)\downarrow$

completeness: $f(x) <_i f(y)\downarrow \implies f(x \ll_i y)\downarrow$

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

morphism : $f(x \ll_i y)\downarrow$ and $f(x) <_i f(y)\downarrow \implies f(x \ll_i y) = f(x) <_i f(y)$

sound: $f(x \ll_i y)\downarrow \implies f(x) <_i f(y)\downarrow$

compleness: $\cancel{f(x) <_i f(y)\downarrow \implies f(x \ll_i y)\downarrow}$

Morphisms

$$(X, \ll, \mathcal{I}) \xrightarrow{f} (Y, <, \mathcal{I})$$

↙ forgets

morphism : $f(x \ll_i y){\downarrow}$ and $f(x) <_i f(y){\downarrow} \implies f(x \ll_i y) = f(x) <_i f(y)$

sound: $f(x \ll_i y){\downarrow} \implies f(x) <_i f(y){\downarrow}$

completeness: $\cancel{f(x) <_i f(y){\downarrow} \implies f(x \ll_i y){\downarrow}}$

Morphisms

more info

forgets

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

morphism: $f(x \ll_i y)\downarrow$ and $f(x) <_i f(y)\downarrow \implies f(x \ll_i y) = f(x) <_i f(y)$

sound: $f(x \ll_i y)\downarrow \implies f(x) <_i f(y)\downarrow$

completeness: $\cancel{f(x) <_i f(y)\downarrow \implies f(x \ll_i y)\downarrow}$

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

more info

forgets

Resolution $R : I \times X \times X \times \{1,2\} \to X$

completeness: $\cancel{f(x) <_i f(y)\downarrow \Rightarrow f(x \ll_i y)\downarrow}$

Morphisms

forgets

more info

$$(X, \ll, I) \xrightarrow{\ f\ } (Y, <, I)$$

Resolution $R : I \times X \times X \times \{1,2\} \to X$

Completeness w.r.t. R : $f(x) <_i f(y) \downarrow \implies \begin{cases} f \cdot R(i, x, y, 1) = fx \\ f \cdot R(i, x, y, 2) = fy \\ f\big(R(-,1) \ll_i R(-,2)\big)\downarrow \end{cases}$

Morphisms

$$(X, \ll, I) \xrightarrow{f} (Y, <, I)$$

THM 2

if f is sound and complete (w.r.t. R)
and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

Morphisms

THM 2

if f is sound and complete (w.r.t. R)

and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

Morphisms

THM 2

if f is sound and complete (w.r.t. R)
and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

forget

Morphisms

THM 2

if f is sound and complete (w.r.t. R)

and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

forget

elaborated terms

Morphisms

THM 2

if f is sound and complete (w.r.t. R)
and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

forget

elaborated terms

free term from which i can extract an elaborated term

Morphisms

THM 2

if f is sound and complete (w.r.t. R)
and R maps $gen_A B$ to $gen_A B$

$$f(gen_A(B)) = gen_{f(A)}(f(B))$$

forget

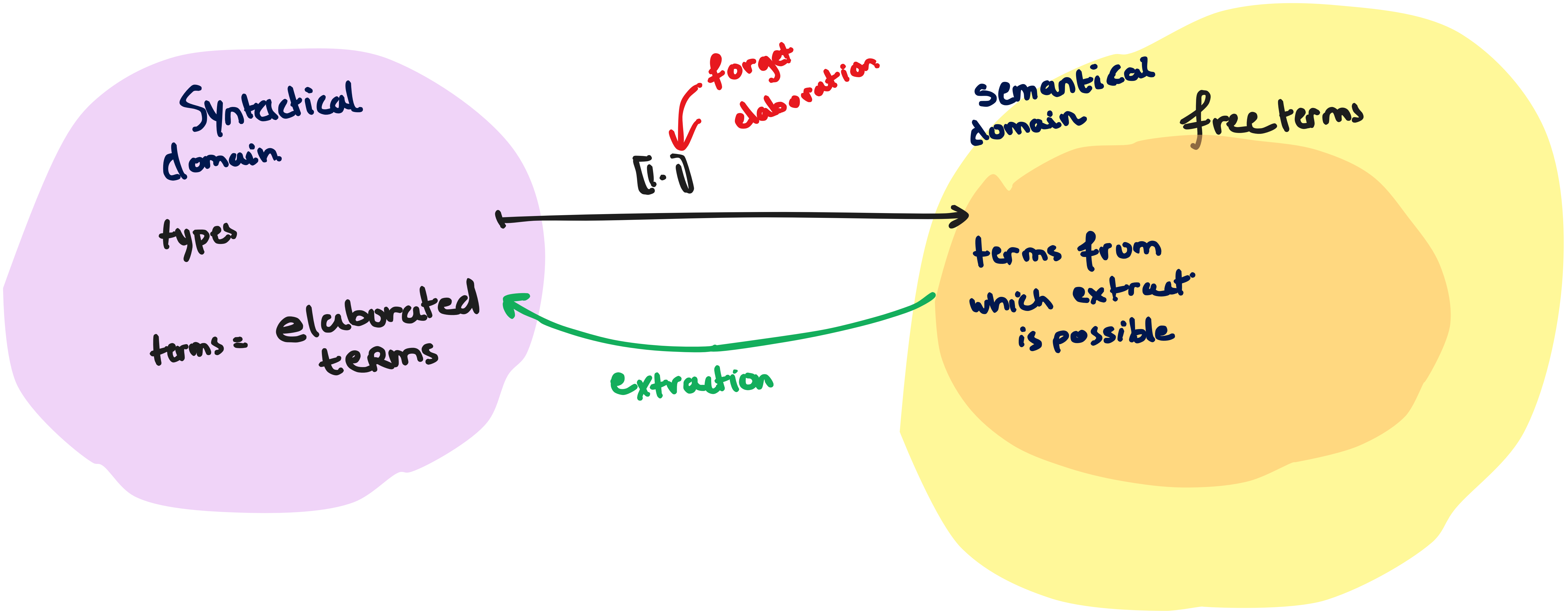
elaborated terms

free term from which i can extract an elaborated term

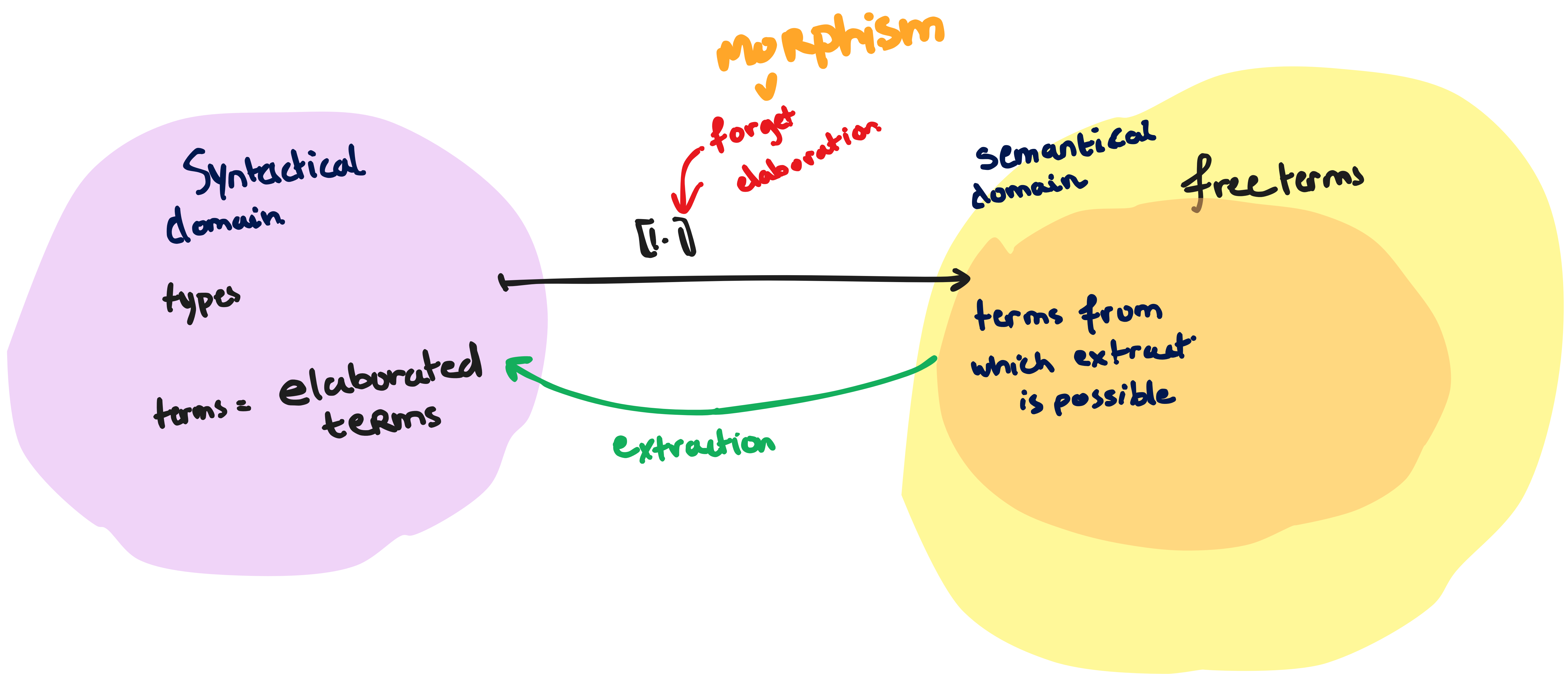
char by coalescence

$\longrightarrow f(B)$

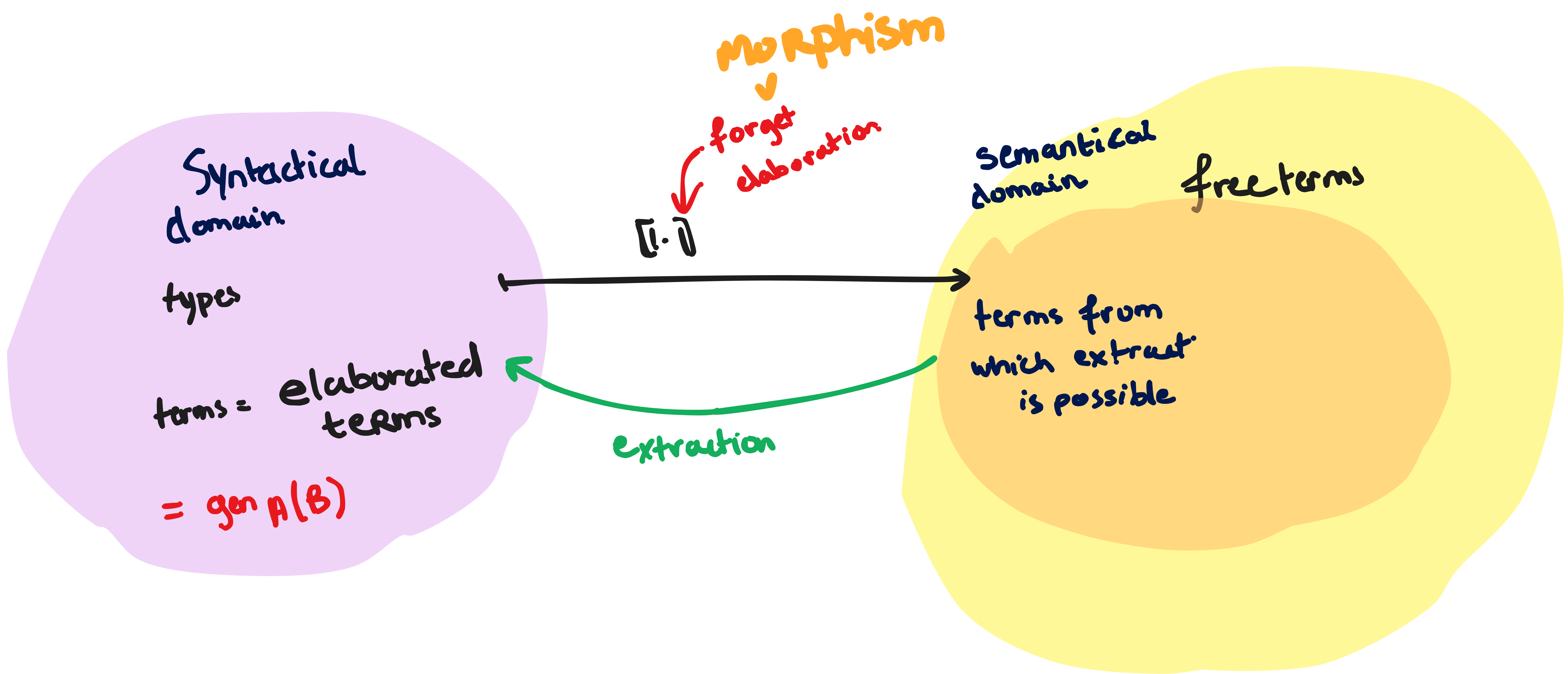
Partial indexed Magma



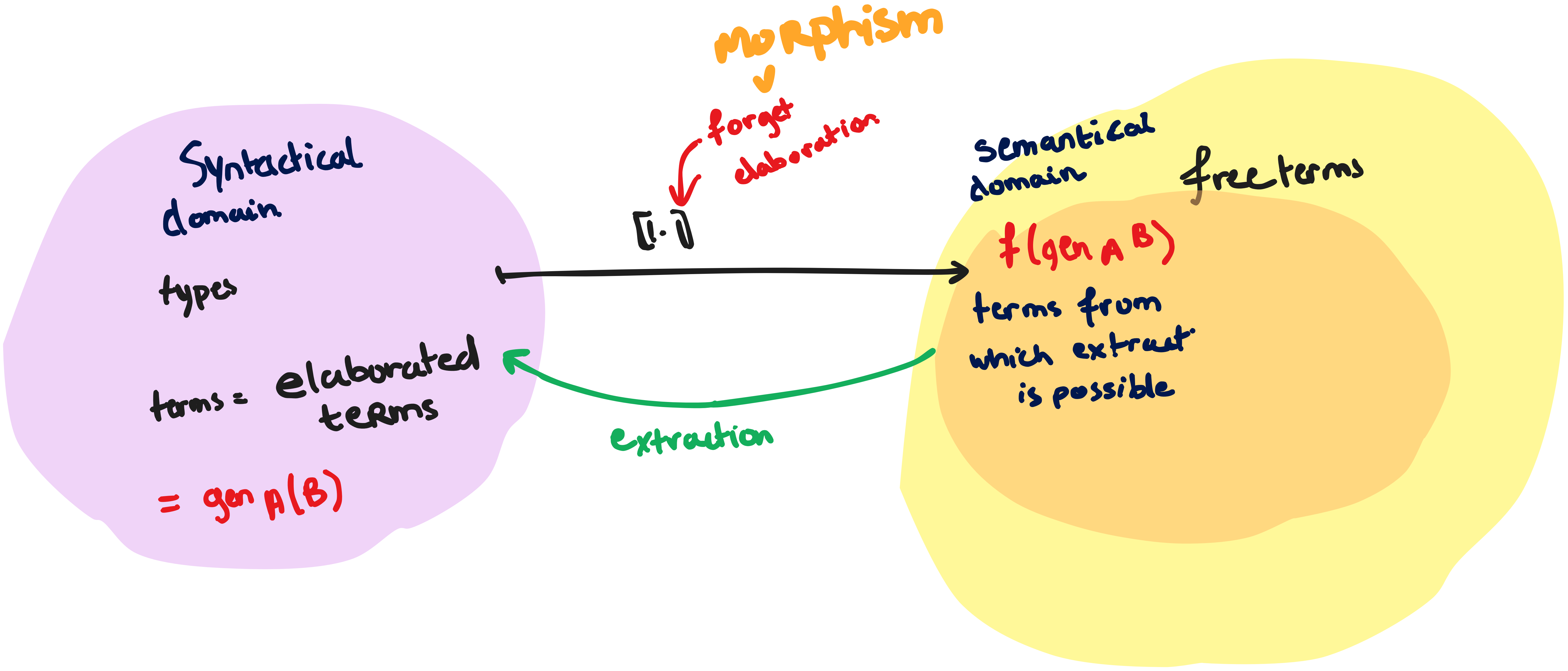
Partial indexed Magma



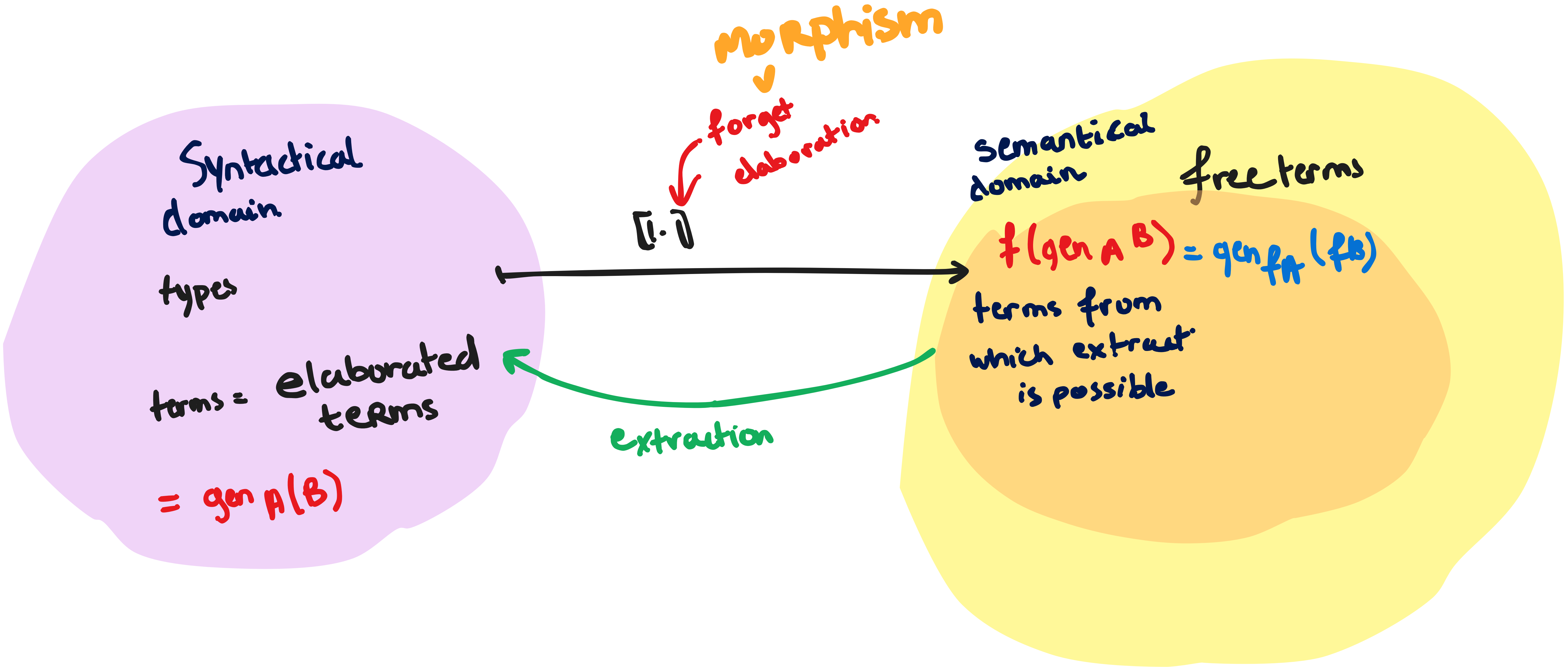
Partial indexed Magma



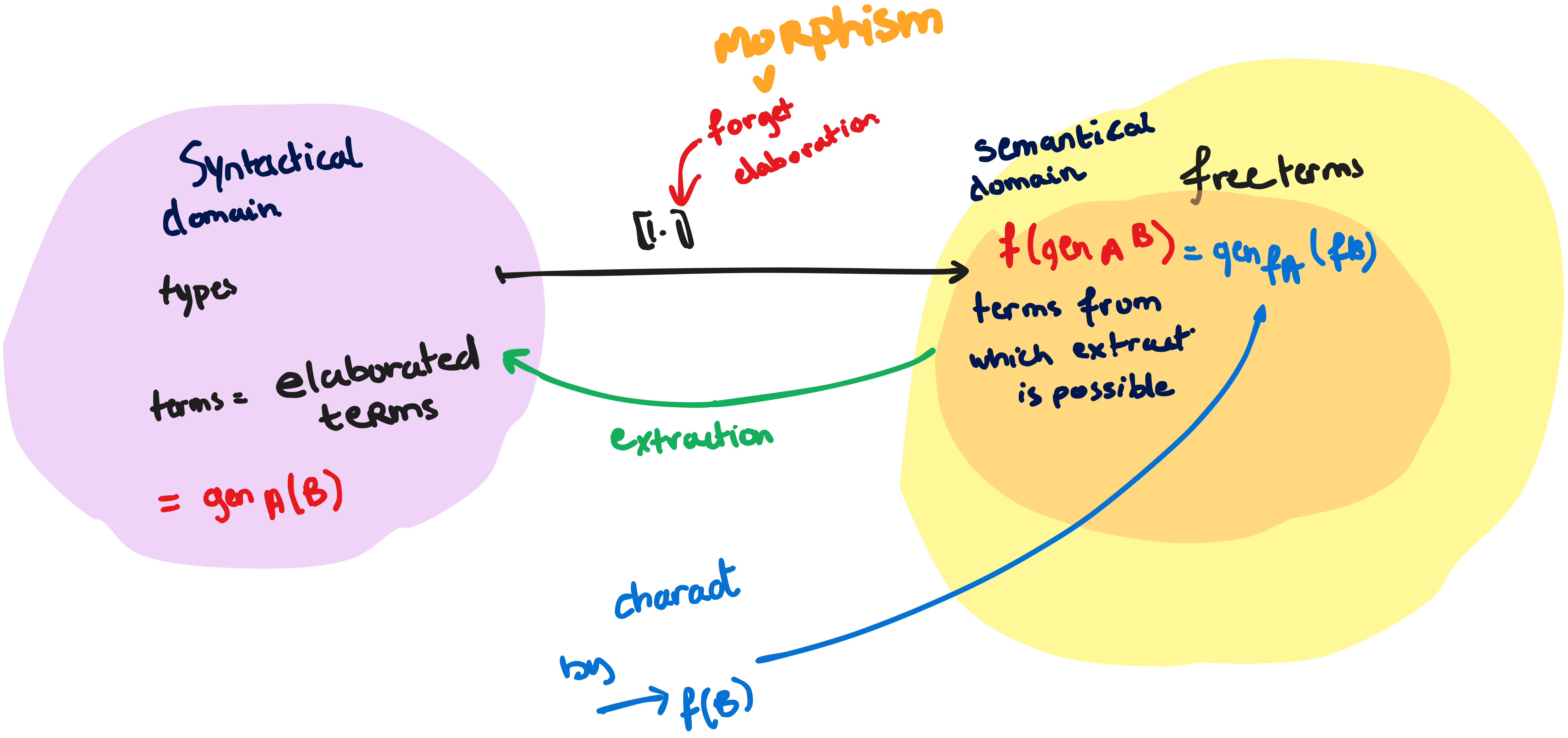
Partial indexed Magma



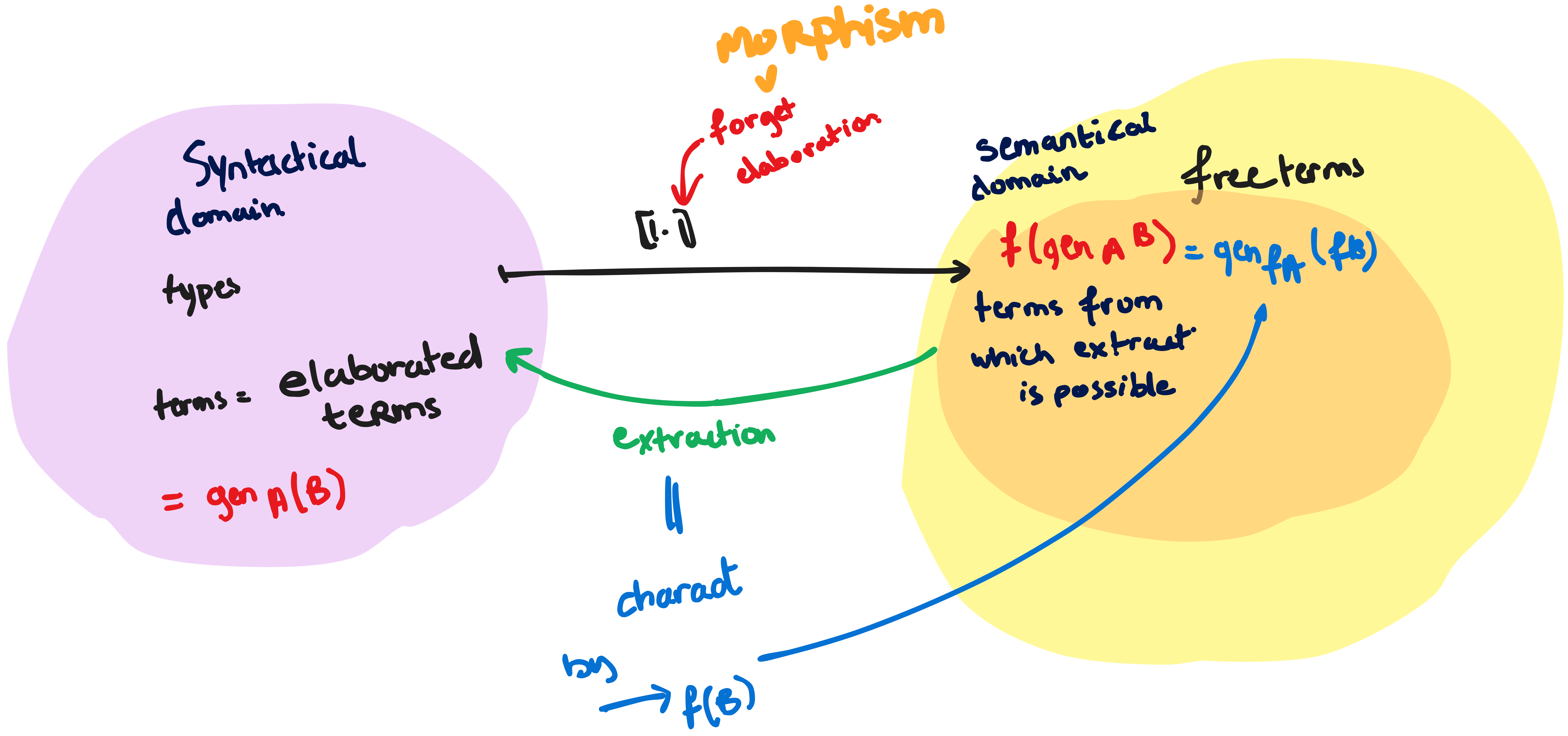
Partial indexed Magma



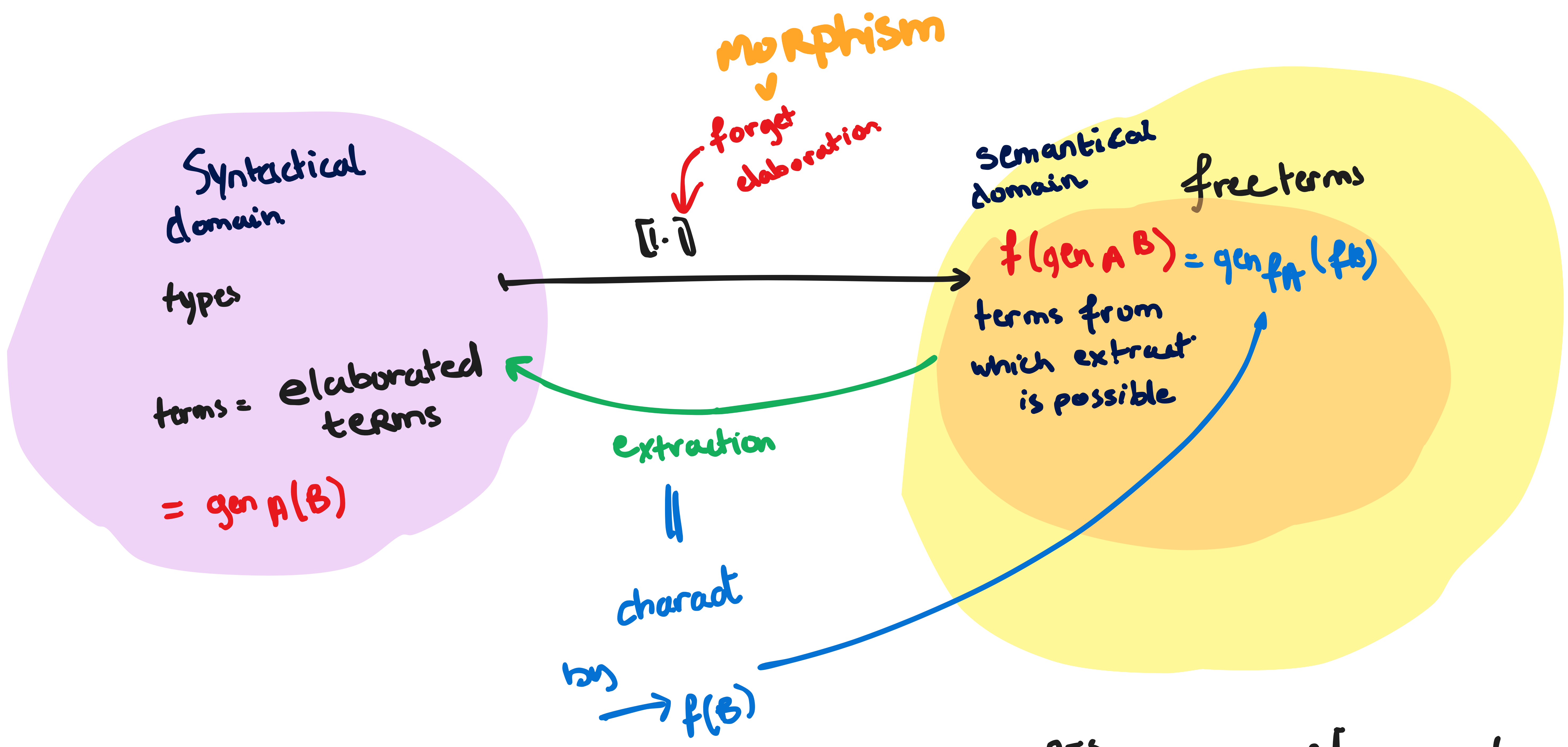
Partial indexed Magma



Partial indexed Magma



Partial indexed Magma



$$x \xrightarrow[f_B]{*} f_a \overset{\text{THM1}}{\Longrightarrow} x = f_a \ll_{i_1} f_{b_1} \ll_{i_2} \cdots \ll_{i_n} f_{b_n} \overset{\text{RES}}{\Longrightarrow} x = f[a' \ll_{i_1} b'_1 \ll_{i_2} \cdots \ll_{i_n} b'_n]$$

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

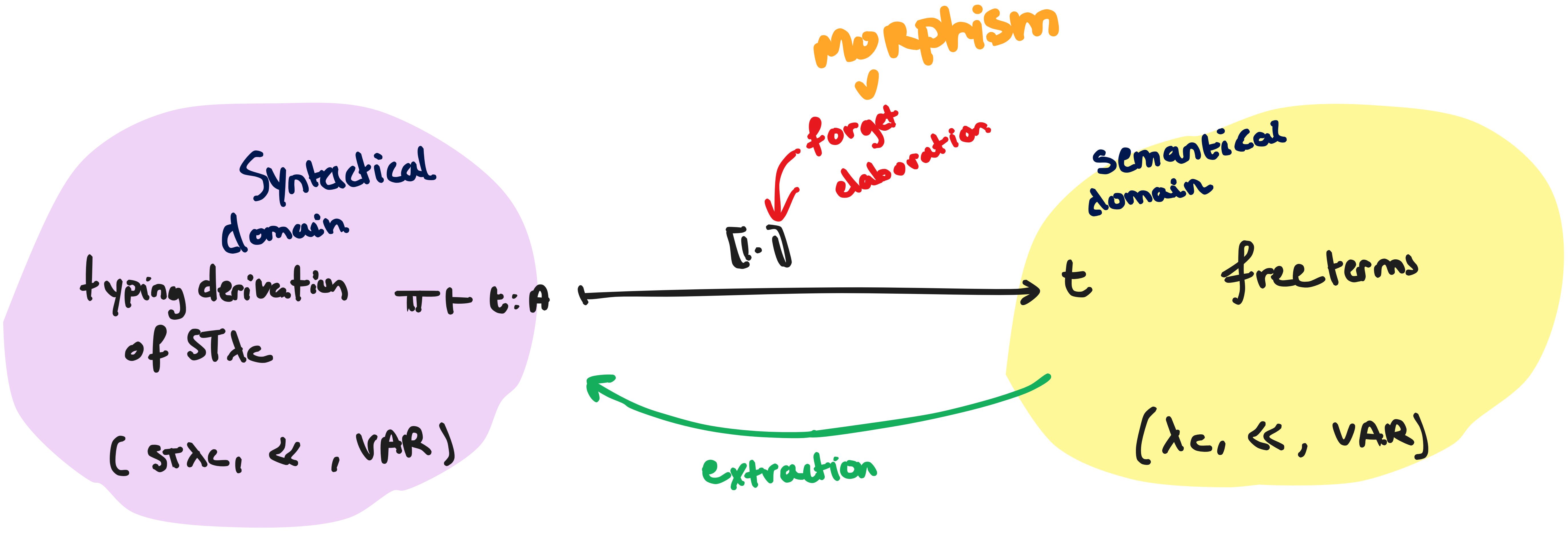
↳ running example

3 - Morphisms

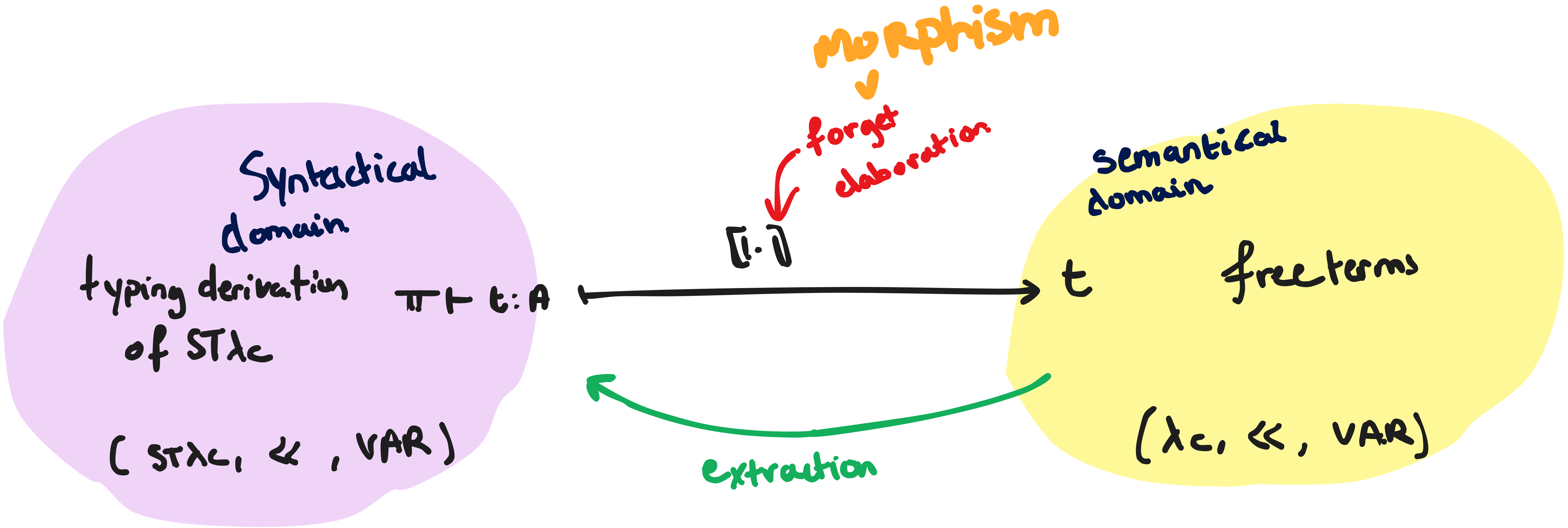
↳ running example

4 - confluence of coalescence

RUNNING EXAMPLE



RUNNING EXAMPLE



$$\dfrac{\dfrac{\Gamma \vdash + : int \to int \to int \qquad \Gamma \vdash x : int \qquad \Gamma \vdash 3 : int}{\Gamma \vdash x + 3 : int}\,\text{tapp}}{\Gamma \vdash \lambda x . x+3 : int \to int}\,\text{abs} \quad \longmapsto \quad \lambda x . x + 3$$

Running Example

$(\mathrm{STX}_c, \ll, \mathrm{VAR})$ as a p.i.m.

RUNNING EXAMPLE

$(\mathrm{ST}\lambda_C, \ll, \mathrm{VAR})$ as a p.i.m.

$$\dfrac{\Gamma \vdash + : int \to int \to int \qquad \Gamma \vdash x : int \qquad \Gamma \vdash 3 : int}{\dfrac{\Gamma \vdash x + 3 : int}{\Gamma \vdash \lambda x. x+3 : int \to int}\ \text{abs}}\ \text{tapp}$$

RUNNING EXAMPLE

$(ST\lambda_C, \ll, VAR)$ as a p.i.m.

$$\dfrac{\dfrac{\Gamma \vdash + : \mathrm{int} \to \mathrm{int} \to \mathrm{int} \qquad \Gamma \vdash x : \mathrm{int} \qquad \Gamma \vdash 3 : \mathrm{int}}{\Gamma \vdash x + 3 : \mathrm{int}}\,\text{+app}}{\Gamma \vdash \lambda x. x+3 : \mathrm{int} \to \mathrm{int}}\,\text{abs}$$

$$= \quad \dfrac{}{\Gamma \vdash e_0 : \mathrm{int} \to \mathrm{int}}\,\bullet \quad \ll_{e_0} \quad \dfrac{\dfrac{}{\Gamma \vdash e_1 : \mathrm{int}}\,\bullet}{\Gamma \vdash \lambda x. e_1 : \mathrm{int} \to \mathrm{int}}\,\text{abs}$$

$$\ll_{e_1} \quad \dfrac{\dfrac{}{\Gamma \vdash + : \mathrm{int}^2 \to \mathrm{int}}\,\text{ax} \qquad \dfrac{}{\Gamma \vdash e_2 : \mathrm{int}}\,\cdot \qquad \dfrac{}{\Gamma \vdash e_3 : \mathrm{int}}\,\cdot}{\Gamma \vdash e_2 + e_3 : \mathrm{int}}\,\text{app-+}$$

$$\ll_{e_2} \quad \dfrac{}{\Gamma \vdash x : \mathrm{int}}\,\text{ax}$$

$$\ll_{e_3} \quad \dfrac{}{\Gamma \vdash 3 : \mathrm{int}}\,\text{ax}$$

RUNNING EXAMPLE

$(\mathrm{ST}\lambda c, \ll, \mathrm{VAR}) \longrightarrow (\lambda c, \ll, \mathrm{Var})$ is sound ?

RUNNING EXAMPLE

$(\mathrm{ST}\lambda c, \ll, \mathrm{VAR}) \longrightarrow (\lambda c, \ll, \mathrm{Var})$ is sound ? YES

RUNNING EXAMPLE

$(\mathrm{STλc}, \ll, \mathrm{VAR}) \longrightarrow (λc, \ll, \mathrm{Var})$ is sound ? YES

$$F\left(\frac{}{\Gamma \vdash e : int \to int} \ll_e \frac{\overline{\Gamma' \vdash e_1 : int}}{\Gamma \vdash λx.\, e_1 : int}\,abs\right)$$

RUNNING EXAMPLE

$(STλc, \preccurlyeq, VAR) \longrightarrow (λc, \preccurlyeq, Var)$ is sound ? YES

$$F\left(\frac{}{\Gamma \vdash e : int \to int} \preccurlyeq_e \frac{\overline{\Gamma' \vdash e_1 : int}}{\Gamma \vdash \lambda x. e_1 : int} abs\right)$$

$$\shortparallel$$

$$F\left(\frac{\overline{\Gamma' \vdash e_1 : int}}{\Gamma \vdash \lambda x. e_1 : int} abs\right)$$

RUNNING EXAMPLE

$(STλc, \ll, VAR) \longrightarrow (λc, \ll, Var)$ is sound ? YES

$$F\left(\frac{}{\Gamma \vdash e : int \to int} \ll_e \frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x.\, e_1 : int}\,abs\right)$$

$$\parallel$$

$$F\left(\frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x.\, e_1 : int}\,abs\right)$$

$$\parallel$$

$$\lambda x.\, e_1$$

Running Example

$$(\mathrm{ST}\lambda_c, \ll, \mathrm{VAR}) \longrightarrow (\lambda_c, \ll, \mathrm{Var}) \text{ is sound ? yes}$$

$$F\left(\frac{}{\Gamma \vdash e : \mathrm{int} \to \mathrm{int}} \ll_e \frac{\overline{\Gamma' \vdash e_1 : \mathrm{int}}\,\vdots}{\Gamma \vdash \lambda x.\, e_1 : \mathrm{int}}\,\mathrm{abs} \right)$$

$$\parallel$$

$$F\left(\frac{\overline{\Gamma' \vdash e_1 : \mathrm{int}}\,\vdots}{\Gamma \vdash \lambda x.\, e_1 : \mathrm{int}}\,\mathrm{abs} \right)$$

$$\parallel$$

$$\lambda x.\, e_1$$

$$\Longrightarrow \quad e \ll_e \lambda x.\, e_1$$

Running Example

$$(ST\lambda c, \ll, VAR) \longrightarrow (\lambda c, \ll, Var) \text{ is sound? YES}$$

$$F\left(\frac{}{\Gamma \vdash e : int \to int}\right) \ll_e \frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x. e_1 : int}\,abs \qquad F\left(\frac{}{\Gamma \vdash e : int \to int}\right) \ll_f F\left(\frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x. e_1 : int}\,abs\right)$$

$$\Vert \qquad\qquad\qquad\qquad\qquad\qquad \Vert$$

$$F\left(\frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x. e_1 : int}\,abs\right)$$

$$\Vert$$

$$\lambda x. e_1 \qquad \Longleftrightarrow \qquad e \ll_e \lambda x. e_1$$

RUNNING EXAMPLE

$(STλc, \ll, VAR) \longrightarrow (λc, \ll, Var)$ is sound ? YES

$$F\left(\frac{}{\Gamma \vdash e : int \to int} \ll_e \frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x.\, e_1 : int}\,\text{abs}\right) = F\left(\frac{}{\Gamma \vdash e : int \to int}\right) \ll_e F\left(\frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x.\, e_1 : int}\,\text{abs}\right)$$

$$\| \qquad\qquad\qquad\qquad\qquad\qquad \|$$

$$F\left(\frac{\overline{\Gamma \vdash e_1 : int}}{\Gamma \vdash \lambda x.\, e_1 : int}\,\text{abs}\right)$$

$$\|$$

$$\lambda x.\, e_1 \qquad\qquad = \qquad\qquad e \ll_e \lambda x.\, e_1$$

RUNNING EXAMPLE

$(\mathrm{ST}\lambda c, \ll, \mathrm{VAR}) \longrightarrow (\lambda c, \ll, \mathrm{Var})$ is complete?

RUNNING EXAMPLE

$(\mathrm{STLC}, \ll, \mathrm{VAR}) \longrightarrow (\lambda c, \ll, \mathrm{Var})$ is complete? careful!

RUNNING EXAMPLE

$(\mathrm{STλ_C}, \ll, \mathrm{VAR}) \longrightarrow (\lambda_c, \ll, \mathrm{Var})$ is complete? carefull

in λ_c :

RUNNING EXAMPLE

$(\mathrm{STλc}, \ll, \mathrm{VAR}) \longrightarrow (\lambda c, \ll, \mathrm{Var})$ is complete? careful!

in λc : $\quad e \ll_e (e_1)e_2 \ll_{e_1} \times \quad \ll_{e_2} \times$

RUNNING EXAMPLE

$(ST\lambda_c, \ll, VAR) \longrightarrow (\lambda_c, \ll, Var)$ is complete? careful!

in λ_c : $\quad e \ll_e (e_1)e_2 \ll_{e_1} \times \quad \ll_{e_2} \times$

$\quad = (e_1)e_2 \ll_{e_1} \times \quad \ll_{e_2} \times$

RUNNING EXAMPLE

$(\mathrm{ST}\lambda_C, \ll, \mathrm{VAR}) \longrightarrow (\lambda_C, \ll, \mathrm{Var})$ is complete? careful!

in λ_C :

$$
\begin{aligned}
& e \ll_e (e_1)e_2 \ll_{e_1} x \quad \ll_{e_2} x \\
= \; & (e_1)e_2 \ll_{e_1} x \quad \ll_{e_2} x \\
= \; & (x)e_2 \quad \ll_{e_2} x
\end{aligned}
$$

RUNNING EXAMPLE

$(ST\lambda_c, \ll, VAR) \longrightarrow (\lambda_c, \ll, Var)$ is complete? careful!

in λ_c : $\quad e \ll_e (e_1)e_2 \ll_{e_1} x \quad \ll_{e_2} x$

$= (e_1)e_2 \ll_{e_1} x \quad \ll_{e_2} x$

$= (x)\, e_2 \qquad \ll_{e_2} x$

$= (x)\, x$

RUNNING EXAMPLE

$(ST\lambda_c, \ll, VAR) \longrightarrow (\lambda_c, \ll, var)$ is complete? carefull

in λ_c : $e \ll_e (e_1)e_2 \ll_{e_1} x \ll_{e_2} x$

$= (e_1)e_2 \ll_{e_1} x \ll_{e_2} x$

$= (x)e_2 \ll_{e_2} x$

$= (x)x$ ← not STλ

Running Example

$(STX_c, \ll, VAR) \longrightarrow (\lambda_c, \ll, Var)$ is complete? carefull

in λ_c : $e \ll_e (e_1)e_2 \ll_{e_1} x \ll_{e_2} x$

$= (e_1)e_2 \ll_{e_1} x \ll_{e_2} x$

$= (x)e_2 \qquad \ll_{e_2} x$ restrict further

$= (x)x$ $\leftarrow$ not STX

RUNNING EXAMPLE

$(\mathrm{ST}\lambda_c, \ll, \mathrm{VAR}) \longrightarrow (\lambda_c, \ll, \mathrm{Var})$ is complete? carefull

in λ_c :

$$e \ll_e (e_1)e_2 \ll_{e1} x \quad \ll_{e2} x$$

$$= (e_1)e_2 \ll_{e1} x \quad \ll_{e2} x$$

$$= (x)\, e_2 \qquad \ll_{e2} x$$

restrict further

$$= (x)\, x$$ ← not STλ

$$\dfrac{\dfrac{}{\Gamma \vdash x : \mathrm{int} \to \mathrm{int}} \qquad \dfrac{}{\Gamma \vdash x : \mathrm{int}}}{\Gamma \vdash (x)\, x : \mathrm{int}}\ \mathrm{app}$$

no such Γ!

Running Example

$(ST\lambda_c, \ll, VAR) \longrightarrow (\lambda_c, \ll, Var)$ is complete? carefull

in λ_c :

$$e \ll_e (e_1)e_2 \ll_{e_1} x \ll_{e_2} x$$
$$= (e_1)e_2 \ll_{e_1} x \ll_{e_2} x$$
$$= (x)\, e_2 \qquad \ll_{e_2} x$$

restrict further

$$= (x)\, x$$

← not STλ

$$\dfrac{\overline{\Gamma \vdash x : int \to int} \qquad \overline{\Gamma \vdash x : int}}{\Gamma \vdash (x)\, x : int}\ \text{app}$$

no such Γ!

actually instead of λ_c

work in $\lambda_c \times Unif$

II Indexed Partial Magmas

1 - main results

2 - partial indexed algebra

↳ running example

3 - Morphisms

↳ running example

4 - confluence of coalescence

Confluence

Confluence

$(X, \ll, I, \sim, \equiv)$

Confluence

congruence $X \times X \longrightarrow \{0,1\}$

$$(X, \ll, I, \sim, \equiv)$$

Confluence

congruence $X \times X \longrightarrow \{0,1\}$

$$(X, \ll, I, \sim, \equiv)$$

renaming $I \times I \to \mathrm{Func}(X, X)$

$(i, j) \longrightarrow [i \to j]$

Confluence

congruence $X \times X \longrightarrow \{0,1\}$

$$(X, \ll, I, \sim, \equiv)$$

renaming $I \times I \to \mathrm{Fun}(X, X)$

$(i, i) \to [i \to j]$

DEF 1 – $B \subseteq X$ is 2-cancellative

$x \ll_i b_1 \ll_j b_2 = x' \ll_n b_2' \ll_m b_1'$, $b_1 \sim b_1'$, $b_2 \sim b_2'$ $\Longrightarrow x \equiv x'$

Confluence

congruence $X \times X \longrightarrow \{0,1\}$

$$(X, \ll, I, \sim, \equiv)$$

renaming $I \times I \rightarrow \mathrm{Fun}(X, X)$

$(i, j) \rightarrow [i \to j]$

DEF1 – $B \subseteq X$ is 2-cancellative

$x \ll_i b_1 \ll_j b_2 = x' \ll_n b_2' \ll_m b_1'$, $b_1 \sim b_1'$, $b_2 \sim b_2'$ $\Longrightarrow$ $X \equiv X'$

DEF2 – $B \subseteq X$ is right-explicit on U

$U := x \ll_i a = x' \ll_j b$ in U

then $\exists x_0 \in X$, $b' \sim b$, $a \sim b'$ s.t. $U = x_0 \ll_n a' \ll_m b'$

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Func}(X, X)$
$(i,j) \to [i \to j]$

DEF 1 – $B \subseteq X$ is 2-cancellative

$x \ll_i b_1 \ll_j b_2 = x' \ll_n b_2' \ll_m b_1'$, $b_1 \sim b_1'$, $b_2 \sim b_2'$ $\Longrightarrow X \equiv X'$

DEF 2 – $B \subseteq X$ is right-explicit on U

$U = x \ll_j a = x' \ll_j b$ in U

then $\exists x_0 \in X$, $b' \sim b$, $a \sim b'$ s.t. $U = x_0 \ll_n a' \ll_m b'$

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Func}(X, X)$

$(i,j) \to [i \to j]$

DEF 1 – $B \subseteq X$ is 2-cancellative

$x \ll_i b_1 \ll_j b_2 = x' \ll_n b_2' \ll_m b_1'$, $b_1 \sim b_1'$, $b_2 \sim b_2'$ $\Longrightarrow x \equiv x'$

DEF 2 – $B \subseteq X$ is right-explicit on U

U: $x \ll_i a = x' \ll_j b$ in U

then $\exists x_0 \in X$, $b' \sim b$, $a \sim b'$ s.t. $U = x_0 \ll_n a' \ll_m b'$

THM 3

$\to$ B is confluent on $\mathcal{U}$ (1:1)

$\Longleftrightarrow$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Fun}(X, X)$

$(i,j) \to [i \to j]$

DEF 1 – $B \subseteq X$ is 2-cancellative

$$x \ll_i b_1 \ll_j b_2 = x' \ll_n b_2' \ll_m b_1' \, , \quad b_1 \sim b_1' \, , \; b_2 \sim b_2' \implies x \equiv x'$$

DEF 2 – $B \subseteq X$ is right-explicit on U

U: $x \ll_j a = x' \ll_j b$ in U

then $\exists x_0 \in X$, $b' \sim b$, $a \sim b'$ s.t. $U = x_0 \ll_n a' \ll_m b'$

THM 3

$\rightarrow B$ is confluent on $\mathcal{U}$ $(1:1)$

$\Longleftrightarrow$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

$\mathcal{U} = \mathrm{gen}_A(B)$

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Func}(X, X)$

$(i,j) \to [i \to j]$

$\mathcal{U} = \mathrm{gen}_A(B)$

THM 3

$\to$ B is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Func}(X, X)$

$(i, j) \to [i \to j]$

THM 3

$\mathcal{U} = \mathrm{gen}_A(B)$

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\Longleftrightarrow$

B is 2. cancellative and right-explicit on $\mathcal{U}$.

---

in practical cases:
(proofnet)

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Func}(X, X)$

$(i, j) \to [i \to j]$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\Longleftrightarrow$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

$\mathcal{U} = \mathrm{gen}_A(B)$

---

in practical cases: 1 – $\to_B$ is SN (and take lin. # steps)
(proofnet)

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Fun}(X,X)$

$(i,i) \to [i \to j]$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\Longleftrightarrow$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

$\mathcal{U} = \mathrm{gen}_A(B)$

---

in practical cases: 1 - $\to_B$ is SN (and take lin. # steps)

(proofnet) 2 - $\to_B$ is 1:1 confluent on $\mathrm{gen}_A\, B$

Confluence

$(X, \ll, I, \sim, \equiv)$

congruence $X \times X \longrightarrow \{0,1\}$

renaming $I \times I \to \mathrm{Func}(X, X)$

$(i, i) \to [i \to j]$

$\mathcal{U} = gen_A(B)$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\Longleftrightarrow$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

---

in practical cases: (proofnet)

1 - $\to_B$ is SN (and take lin. # steps)

2 - $\to_B$ is 1:1 confluent on $gen_A B$

$\Rightarrow$ quadratic-time extraction / elaboration.

by trying $x \to^* a$, $a \in A$

I Overview

II Indexed Partial Magmas

III Conclusion

MAIN RESULTS

THM 1

$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \to_B a$

MAIN RESULTS

THM 1

$$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\quad x \to_B a$$

?

?

MAIN RESULTS

THM 1

$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \to_B a$

?

MAIN RESULTS

THM 1

$$x \in \mathrm{gen}_A(B) \iff \exists a \in A, \quad x \to_B a$$

MAIN RESULTS

THM 1

$$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \rightarrow_B a$$

THM 2

if f is sound and complete (w.r.t. R)
and R maps $\mathrm{gen}_A B$ to $\mathrm{gen}_A B$

$$f(\mathrm{gen}_A(B)) = \mathrm{gen}_{f(A)}(f(B))$$

MAIN RESULTS

THM 1

$x \in \text{gen}_A(B) \iff \exists a \in A,\ x \to_B a$

THM 2

if f is sound and complete (w.r.t. R)

and R maps $\text{gen}_A B$ to $\text{gen}_A B$

$$f(\text{gen}_A(B)) = \text{gen}_{f(A)}(f(B))$$

?

?

MAIN RESULTS

THM 1

$x \in \text{gen}_A(B) \iff \exists a \in A,\ x \to_B a$

THM 2

if f is sound and complete (w.r.t. R)

?

and R maps $\text{gen}_A B$ to $\text{gen}_A B$

$f(\text{gen}_A(B)) = \text{gen}_{f(A)}(f(B))$

MAIN RESULTS

THM 1

$x \in gen_A(B) \iff \exists a \in A, \; x \rightarrow_B a$

THM 2

if f is sound and complete (w.r.t. R)
and R maps $gen_A B$ to $gen_A B$

$f(gen_A(B)) = gen_{f(A)}(f(B))$

MAIN RESULTS

THM 1

$x \in \text{gen}_A(B) \iff \exists a \in A,\ x \rightarrow_B a$

THM 2

if f is sound and complete (w.r.t. R)
and R maps $\text{gen}_A B$ to $\text{gen}_A B$

$$f(\text{gen}_A(B)) = \text{gen}_{f(A)}(f(B))$$

THM 3

$\rightarrow_B$ is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2. cancellative and right-explicit on $\mathcal{U}$.

MAIN RESULTS

THM 1

$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \to_B a$

THM 2

if f is sound and complete (w.r.t. R)

and R maps $\mathrm{gen}_A B$ to $\mathrm{gen}_A B$

$f(\mathrm{gen}_A(B)) = \mathrm{gen}_{f(A)}(f(B))$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

?

?

MAIN RESULTS

THM 1

$x \in gen_A(B) \iff \exists a \in A, \; x \to_B a$

THM 2

if f is sound and complete (w.r.t. R)

and R maps $gen_A B$ to $gen_A B$

$f(gen_A(B)) = gen_{f(A)}(f(B))$

THM 3

$\to_B$ is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

?

MAIN RESULTS

THM 1

$x \in \mathrm{gen}_A(B) \iff \exists a \in A,\ x \rightarrow_B a$

THM 2

if f is sound and complete (w.r.t. R)

and R maps $\mathrm{gen}_A B$ to $\mathrm{gen}_A B$

$f(\mathrm{gen}_A(B)) = \mathrm{gen}_{f(A)}(f(B))$

THM 3

$\rightarrow_B$ is confluent on $\mathcal{U}$ (1:1)

$\iff$

B is 2-cancellative and right-explicit on $\mathcal{U}$.

Conclusions

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

What's Next?

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

What's Next?

- case 101, proper study of STλC (Hindley-Milner)

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

What's Next?

- case 101, proper study of STλc (Hindley-Milner)
- challenge assertion *
  . . . .

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

What's Next?

- case 101, proper study of STλc (Hindley-Milner)
- challenge assertion *
  - bidirectional types?
  - polymorphic types

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

What's Next?

- case 101, proper study of STλc (Hindley-Milner)
- challenge assertion *
  - bidirectional types?
  - polymorphic types
  - dependant types
  - recursive types
  - ...

CONCLUSIONS

- A rich yet simple algebraic framework : ind. partial magma
- able to model front-end passes in compiler (*)
- applied successfully to proof nets (MLL2)

What's Next?

- case 101, proper study of STλc (Hindley-Milner)
- challenge assertion *
  - bidirectional types ?
  - polymorphic types
  - dependant types
  - recursive types
  - ...

↳ coalescence as a general type-inf algorithm ?

Thank You!