```agda
choice-idempotency : ∀ {i : Size} {A : Domain} {D : Dimension} {e : BCC i A}
    ---------------------------
  → BCC , ⟦_⟧ ⊢ D ⟨ e , e ⟩ ≈ e
choice-idempotency {i} {A} {D} {e} = extensionality (λ c →
  ⟦ D ⟨ e , e ⟩ ⟧ c           ≡⟨⟩
  ⟦ if (c D) then e else e ⟧ c  ≡⟨ Eq.cong (flip ⟦_⟧ c) (if-idemp (c D)) ⟩
  ⟦ e ⟧ c                     ∎)
```

❶ - 8.9k BCC.lagda.md Agda

```agda
completeness-by-expressiveness : ∀ {L₁ L₂ : VarLang} {C₁ C₂ : ConfLang} {S₁ : Se
  → Complete L₁ C₁ S₁
  → L₂ , S₂ is-as-expressive-as L₁ , S₁
    ------------------------------------
  → Complete L₂ C₂ S₂
completeness-by-expressiveness {L₁} {L₂} {_} {_} {S₁} {S₂} encode-in-L₁ L₁-to-L₂
  let {-
```

❷ - 11k Completeness.lagda.md Agda

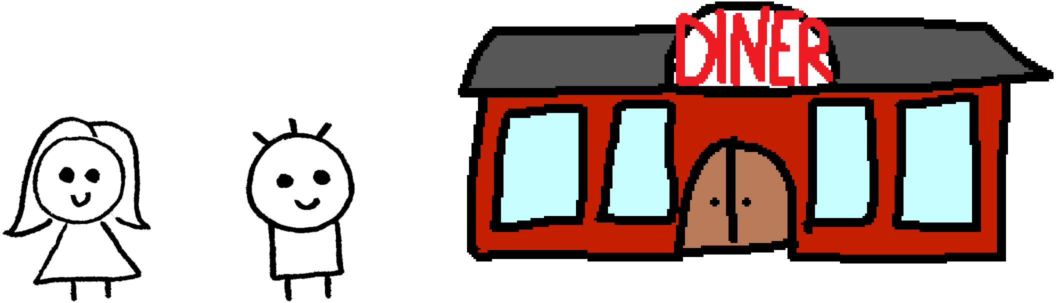
Formal Languages for Solution-Space Variability

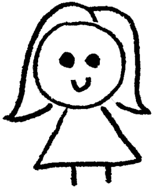
Paul Bittner, Jeffrey Young, Parisa Ataei, Alexander Schultheiß, Eric Walkingshaw, Leopoldo Teixeira, Thomas Thüm | FOSD 2023

Software Engineering
Programming Languages

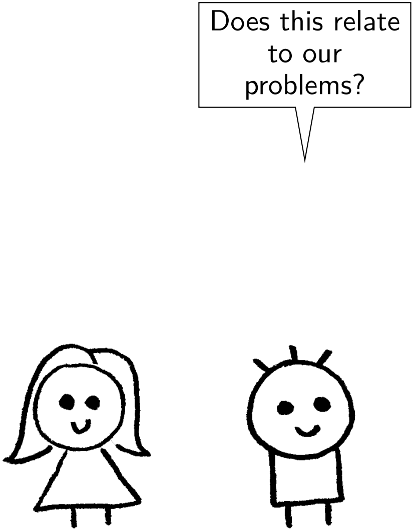
SoftVarE

universität ulm


DINER



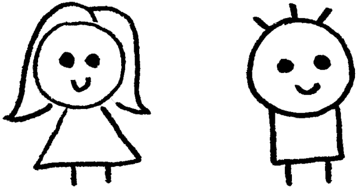

VEGETARIAN
WHICH WICH WOULD YOU LIKE?
TRIPLE CHEESE MELT
ELVIS WICH
TOMATO & AVOCADO
BLACK BEAN PATTY
HUMMUS & BELL PEPPERS
CHOOSE YOUR BREAD
WHITE
WHEAT
CHOOSE YOUR CHEESE (Optional)
AMERICAN
CHEDDAR
SWISS
PEPPER JACK
PROVOLONE
MOZZARELLA
How Would You Like Your WICH Worked?
MUSTARDS
Yellow
Dijon
Honey
Deli
MAYOS
Regular
Lite
Horseradish
Spicy
SPREADS & SAUCES
BBQ
Buffalo
Marinara
1000 Island
Ranch
ONIONS
Red
Grilled
Crispy Strings
VEGGIES
Lettuce
Tomato
Pickles
Jalapenos
Olive Salad
Mushrooms
Sauerkraut
Coleslaw
Bell Peppers
OILS & SPICES
Oil
Vinegar
Parmesan
Oregano

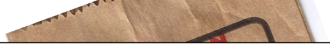

Does this relate to our problems?


VEGETARIAN
WHICH WICH WOULD YOU LIKE?
TRIPLE CHEESE MELT
ELVIS WICH
TOMATO & AVOCADO
BLACK BEAN PATTY
HUMMUS & BELL PEPPERS
CHOOSE YOUR BREAD
WHITE
WHEAT
CHOOSE YOUR CHEESE (Optional)
AMERICAN
SWISS
PROVOLONE
CHEDDAR
PEPPER JACK
MOZZARELLA
How Would You Like Your WICH Worked?
MUSTARDS
Yellow
Dijon
Honey
Deli
MAYOS
Regular
Lite
Horseradish
Spicy
SPREADS & SAUCES
BBQ
Buffalo
Marinara
1000 Island
Ranch
ONIONS
Red
Grilled
Crispy Strings
VEGGIES
Lettuce
Tomato
Pickles
Jalapenos
Olive Salad
Mushrooms
Sauerkraut
Coleslaw
Bell Peppers
OILS & SPICES
Oil
Vinegar
Oregano
Parmesan

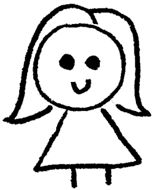
Does this relate to our problems?

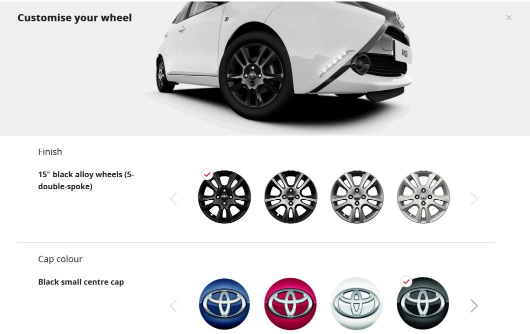
There certainly is a lot of variability in these sandwiches! Just as in ...

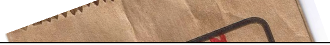




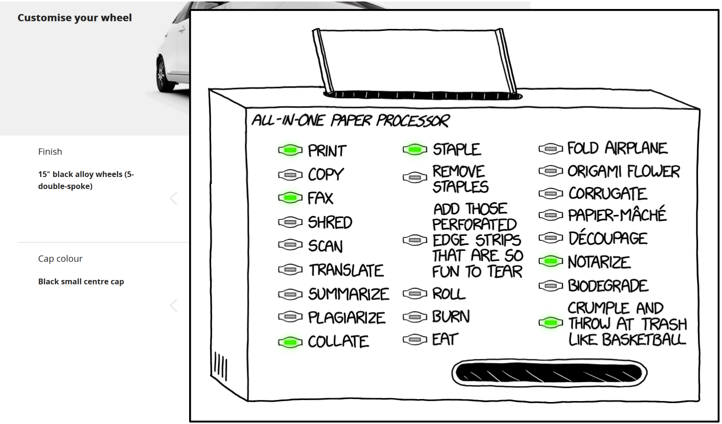
There certainly is a lot of variability in these sandwiches! Just as in …

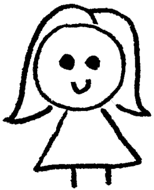




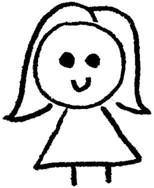


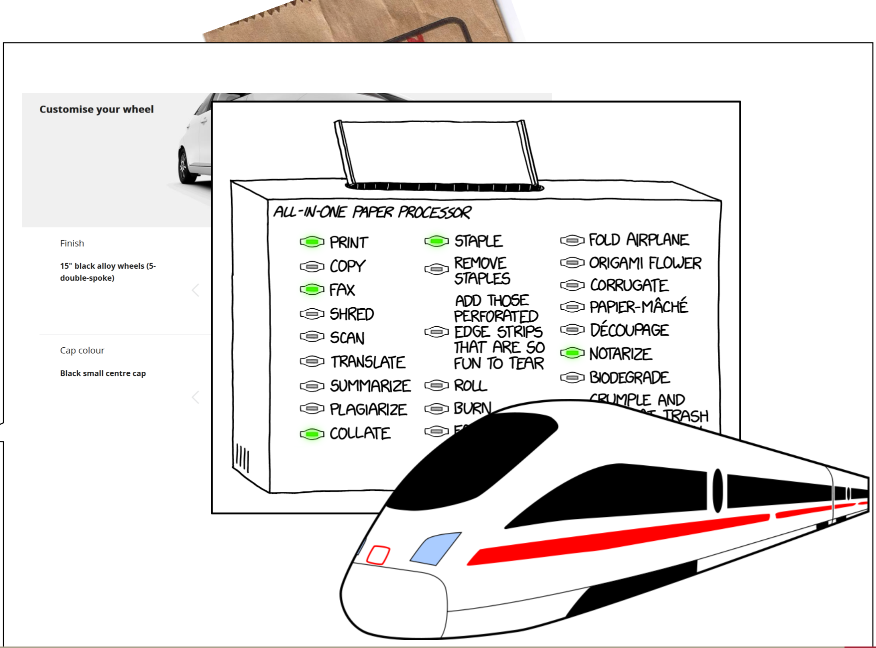
There certainly is a lot of variability in these sandwiches! Just as in ...

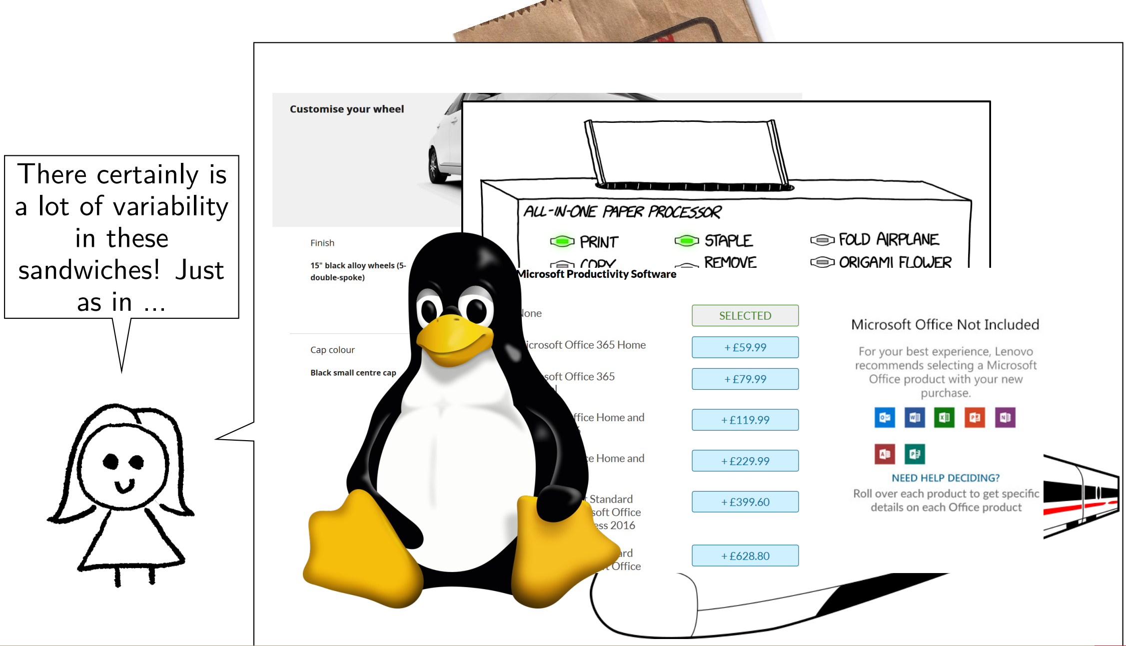


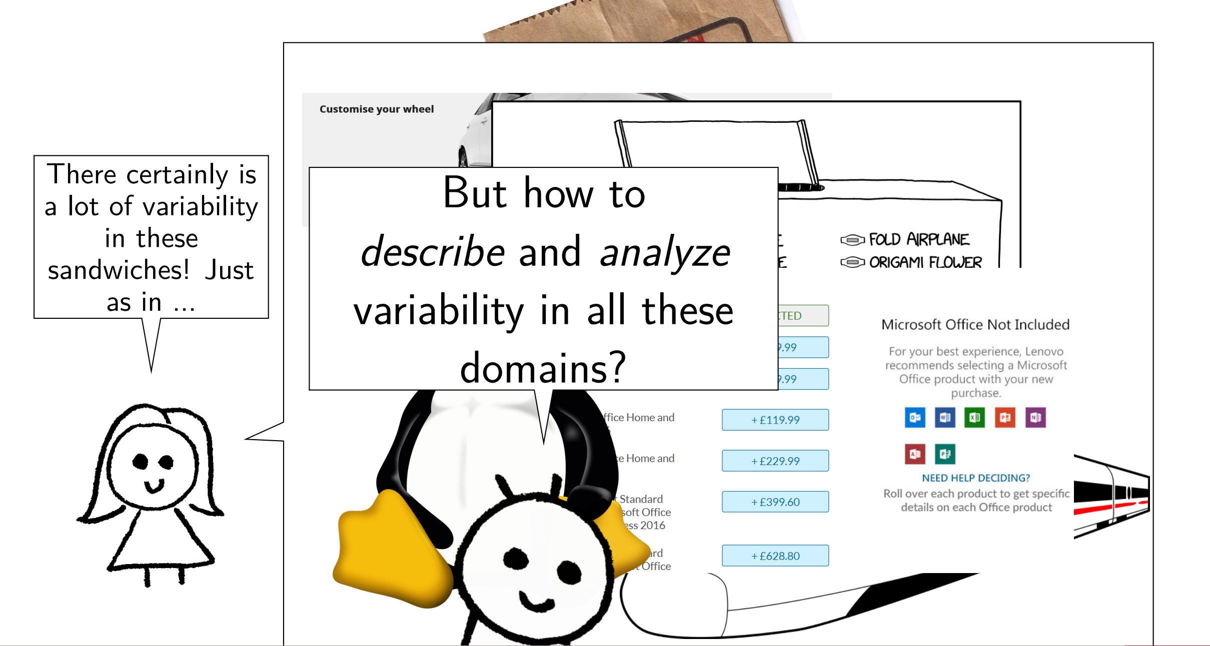


There certainly is a lot of variability in these sandwiches! Just as in ...






There certainly is a lot of variability in these sandwiches! Just as in ...
Customise your wheel
Finish
15" black alloy wheels (5-double-spoke)
Cap colour
Black small centre cap
ALL-IN-ONE PAPER PROCESSOR
PRINT
STAPLE
FOLD AIRPLANE
REMOVE
ORIGAMI FLOWER
Microsoft Productivity Software
SELECTED
+ £59.99
+ £79.99
+ £119.99
+ £229.99
+ £399.60
+ £628.80
Microsoft Office Not Included
For your best experience, Lenovo recommends selecting a Microsoft Office product with your new purchase.
NEED HELP DECIDING?
Roll over each product to get specific details on each Office product


There certainly is a lot of variability in these sandwiches! Just as in ...
But how to *describe* and *analyze* variability in all these domains?
Customise your wheel
FOLD AIRPLANE
ORIGAMI FLOWER
+ £119.99
+ £229.99
+ £399.60
+ £628.80
Microsoft Office Not Included
For your best experience, Lenovo recommends selecting a Microsoft Office product with your new purchase.
NEED HELP DECIDING?
Roll over each product to get specific details on each Office product

Using Formal Languages for Variability

Core
Choice Calculus

Binary
Choice Calculus

Option Calculus

Artifact Trees

Algebraic
Decision Diagrams

Variation Trees

Variability-Aware
Abstract Syn-
tax Trees

Binary
Decision Diagrams

Using Formal Languages for Variability

Core
Choice Calculus

Binary
Choice Calculus

Projectional Editing
[Walkingshaw and Ostermann, 2014]
[Stănciulescu et al., 2016]

Variational Type Inference
[Chen et al., 2014]

Variational Databases
[Ataei et al., 2021]

Variational SAT and SMT Solving
[Young et al., 2022]

...

ees

Algebraic
Decision Diagrams

Variation Trees

Variability-Aware
Abstract Syn-
tax Trees

Binary
Decision Diagrams

Using Formal Languages for Variability

Core
Choice Calculus

Binary
Choice Calculus

Option Calculus

Artifact Trees

Algebraic
Decision Diagrams

Formalizing Software Product-Line Analyses
[Castro et al., 2021]

Classical Computation Problems
[Bahar et al., 1993]

Game Theory
[Aadithya et al., 2011]

. . .

ware
yn-
s

Binary
Decision Diagrams

Using Formal Languages for Variability

Core
Choice Calculus

Binary
Choice Calculus

Option Calculus

Artifact Trees

Syntax-Checking for Variable Software
[Kästner, 2010]

Variability-Aware Parsing
[Kästner et al., 2011]

Feature Trace Recording
[Bittner et al., 2021]

...

Variability-Aware Abstract Syntax Trees

Binary
Decision Diagrams

Using Formal Lang

Core
Choice Calculus

Binary
Choice Calculus

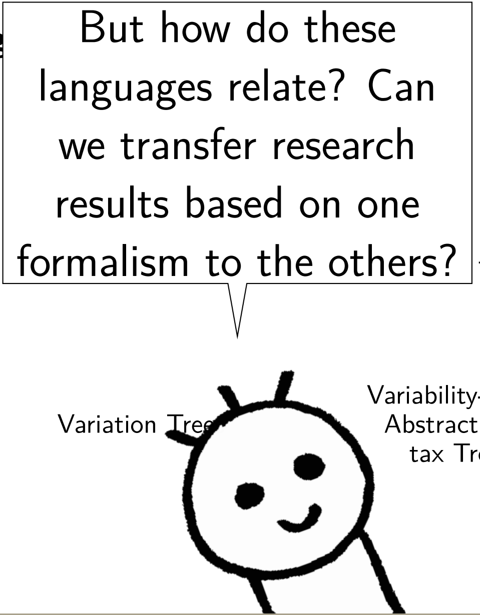
Algebraic
Decision Diagrams

Binary
Decision Diagrams

Trees

Variation Tree

Variability-Aware
Abstract Syn-
tax Trees



Using Formal Lang

But how do these languages relate? Can we transfer research results based on one formalism to the others?

Core
Choice Calculus

Binary
Choice Calculus

Algebraic
Decision Diagrams

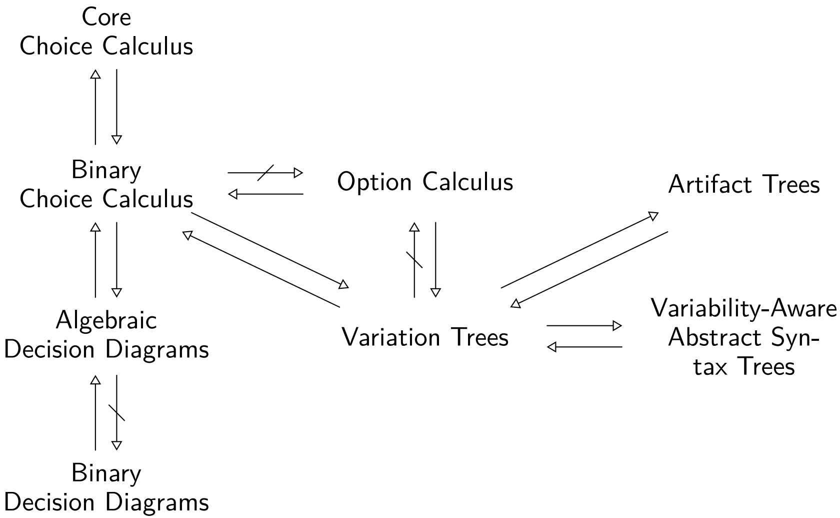
Binary
Decision Diagrams

Trees

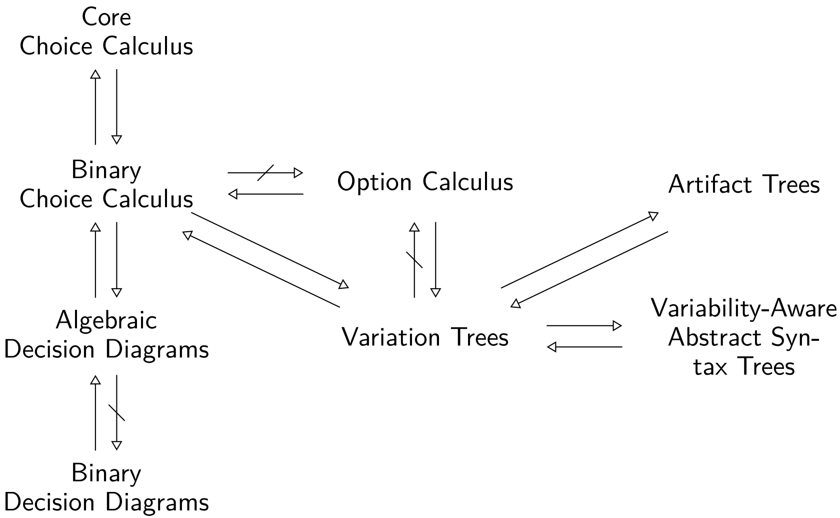
Variation Tree

Variability-Aware
Abstract Syn-
tax Trees

Research Goal Map Out Language Space

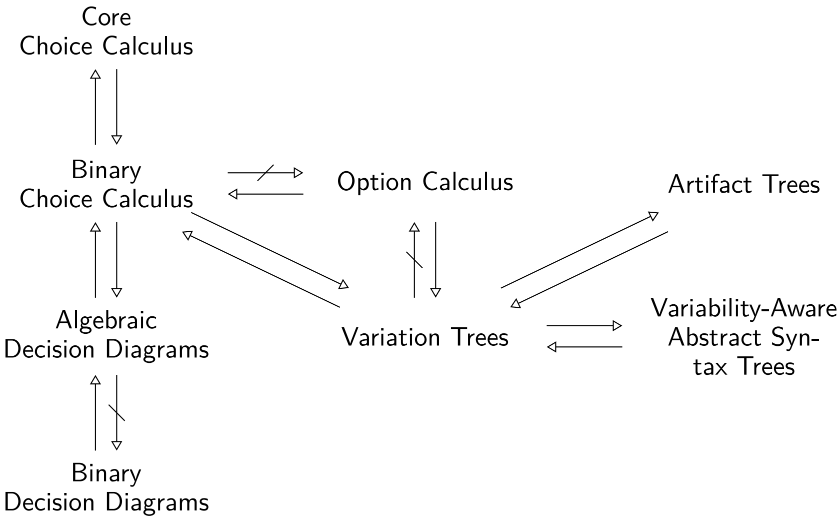


Research Goal Map Out Language Space



Why relating?

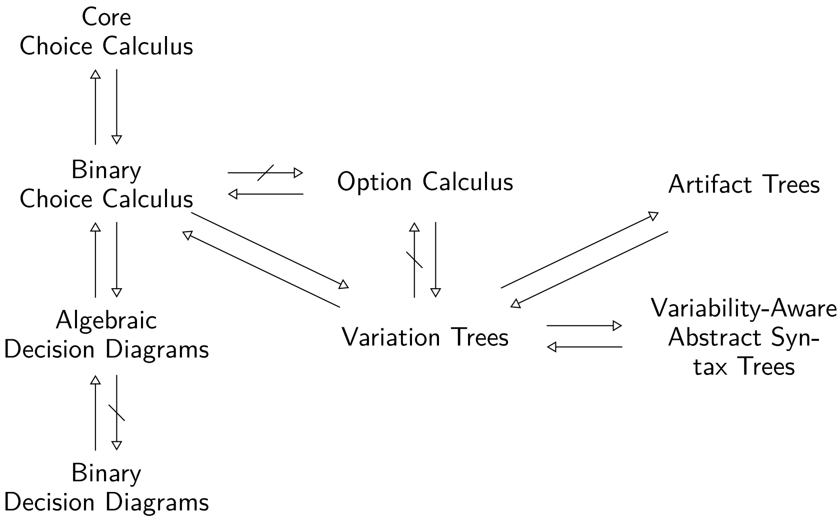
Research Goal Map Out Language Space



Why relating?

transfer research results

Research Goal Map Out Language Space

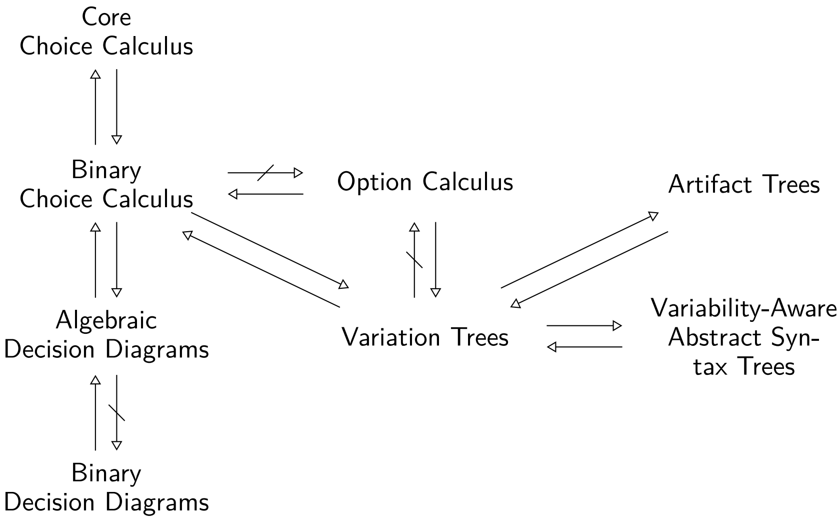


Why relating?

transfer research results

guidance on which language to pick

Research Goal Map Out Language Space



Why relating?

transfer research results

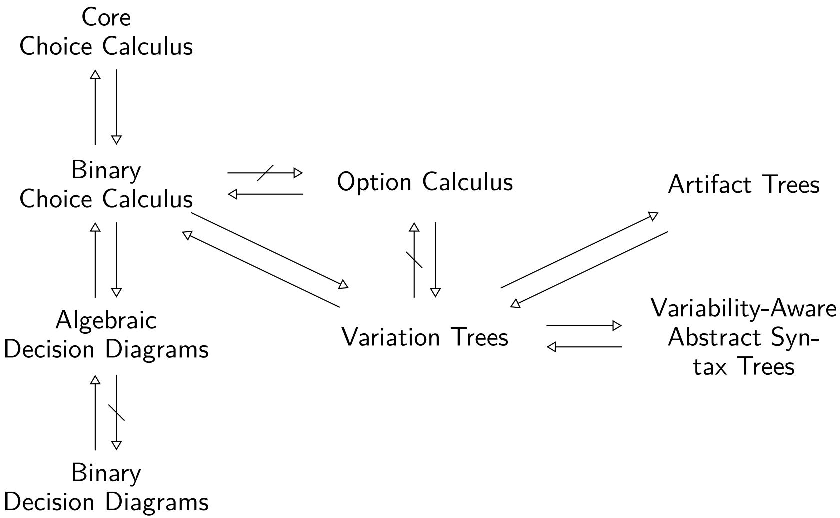
guidance on which language to pick

perform analyses or proofs on "easy"-to-process language

↕

base user-languages on expressive formal languages

Research Goal **Map Out Language Space**



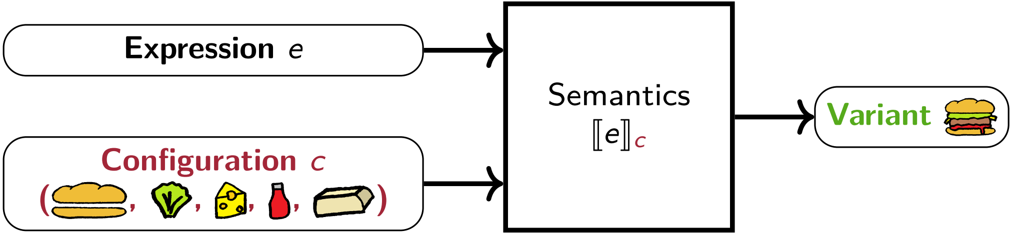
Why relating?

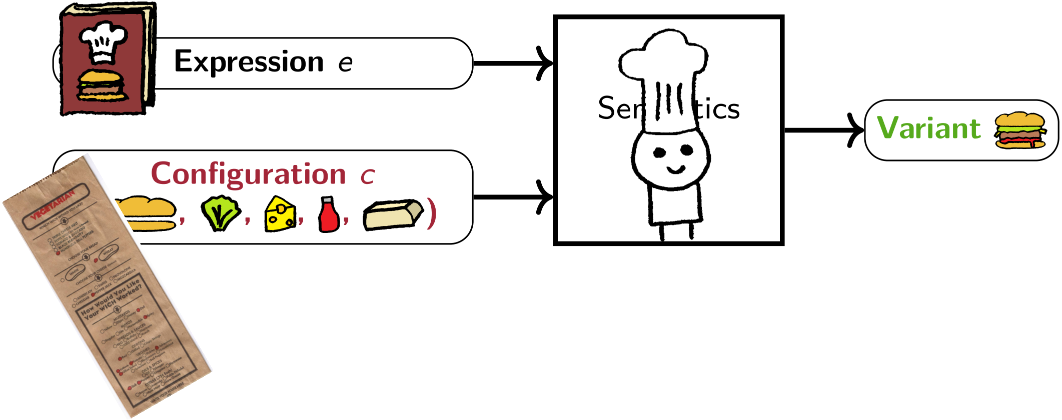
transfer research results

guidance on which language to pick

perform analyses or proofs on "easy"-to-process language
↕
base user-languages on expressive formal languages

a better understanding of variability itself


Expression e
Configuration c
Semantics
$[\![e]\!]_c$
Variant


Expression e
Configuration c
)
Ser
tics
Variant
VEGETARIAN
How Would You Like Your WICH Worked?

Example: Core Choice Calculus [Walkingshaw, 2013]

$$
\begin{array}{lcll}
e & ::= & a\prec e, \ldots, e\succ & \textit{Object Structure} \\
  & | & D\langle e, \ldots, e\rangle & \textit{Choice}
\end{array}
$$

Example: Core Choice Calculus [Walkingshaw, 2013]

$$
\begin{array}{rcll}
e & ::= & a\prec e, \ldots, e \succ & \textit{Object Structure} \\
  & \mid & D\langle e, \ldots, e \rangle & \textit{Choice}
\end{array}
$$

always 

maybe 

always 

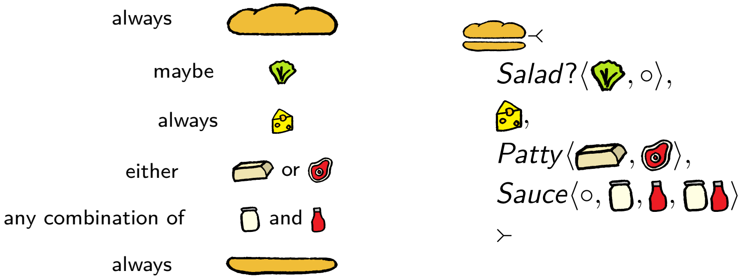
either  or 

any combination of  and 

always 

Example: Core Choice Calculus [Walkingshaw, 2013]

$$
\begin{array}{rcll}
e & ::= & a\prec e, \ldots, e\succ & \textit{Object Structure} \\
 & | & D\langle e, \ldots, e\rangle & \textit{Choice}
\end{array}
$$



Example: Core Choice Calculus [Walkingshaw, 2013]

$$
\begin{array}{rcll}
e & ::= & a\prec e, \ldots, e \succ & \textit{Object Structure} \\
  & \mid & D\langle e, \ldots, e\rangle & \textit{Choice}
\end{array}
$$

always

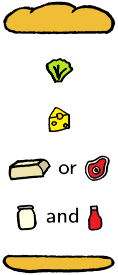
maybe

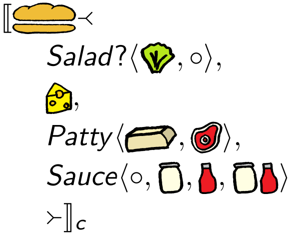
always

either … or

any combination of … and

always



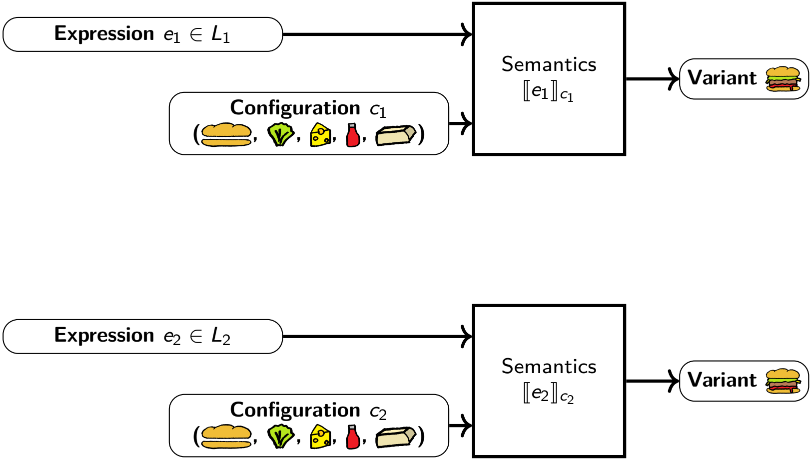


=

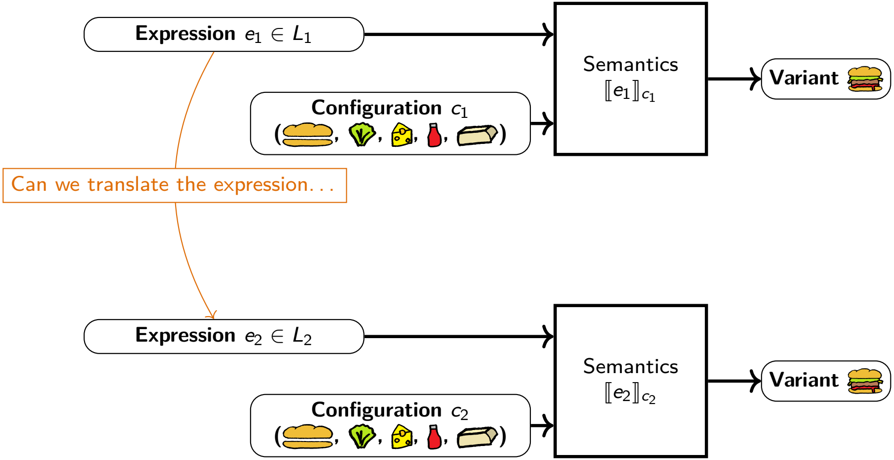
$$
\text{if } \begin{array}{ll} c(\textit{Salad?}) & = 0, \\ c(\textit{Patty}) & = 0, \\ c(\textit{Sauce}) & = 2. \end{array}
$$

How to Compare Variability Languages?

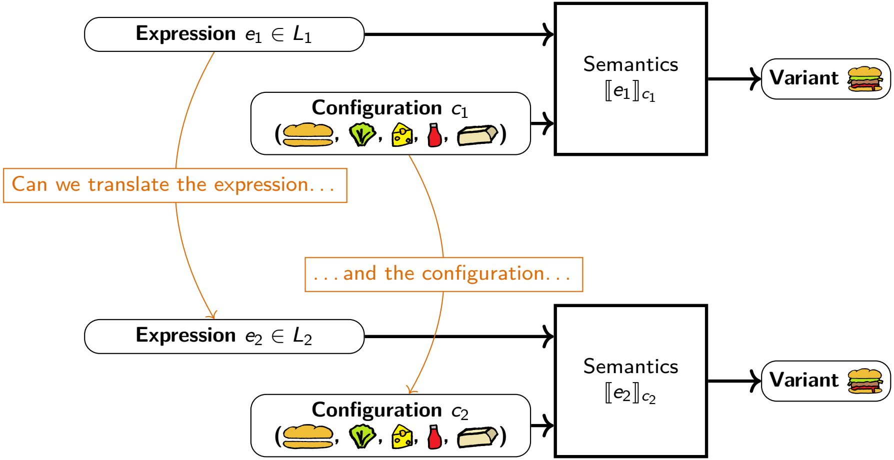
How to Compare Variability Languages?



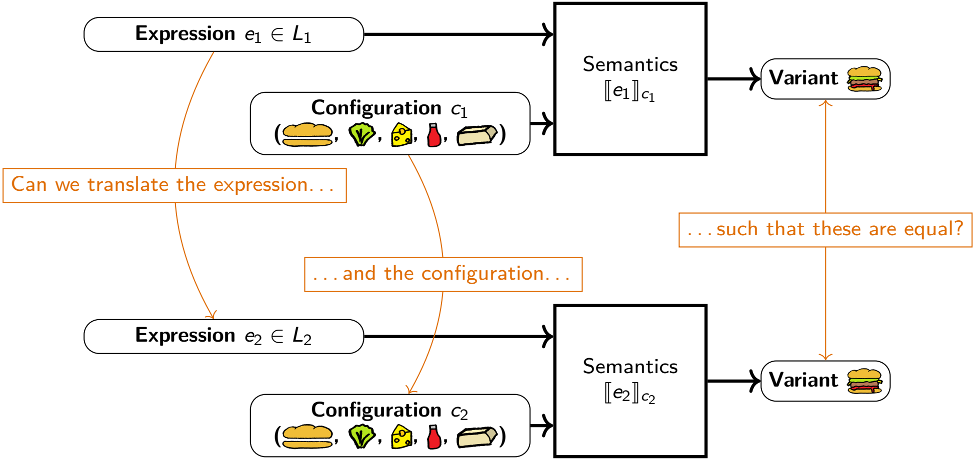
How to Compare Variability Languages?



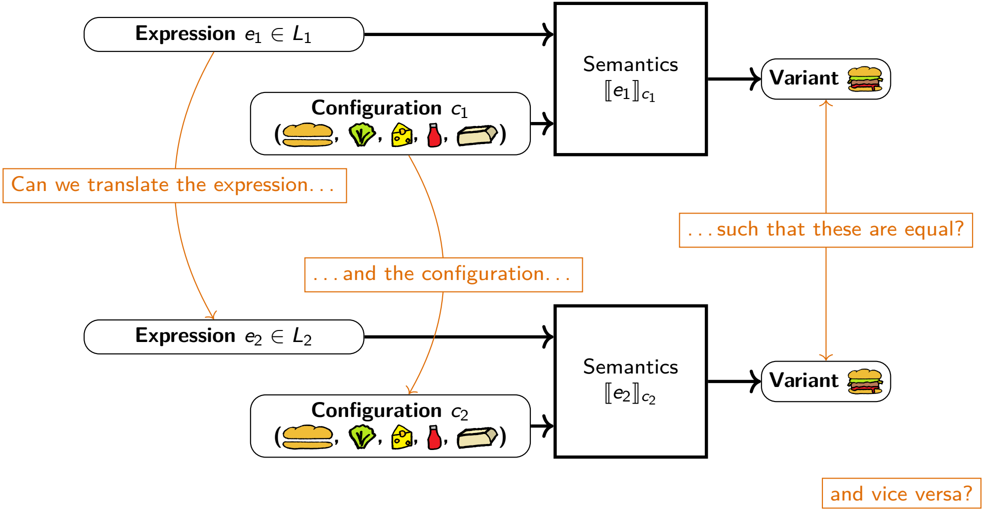
How to Compare Variability Languages?



How to Compare Variability Languages?



How to Compare Variability Languages?

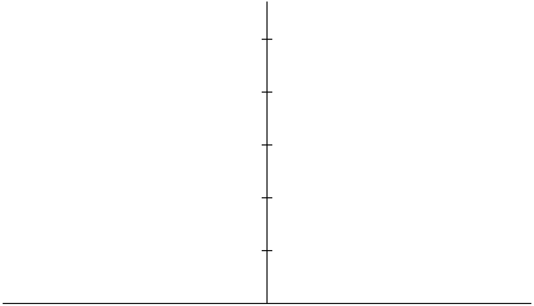


L_1 is as expressive as L_2 **iff** Every expression in L_2 can be translated to an expression in L_1 that describes the same set of variants.

L_1 is as expressive as L_2	**iff**	Every expression in L_2 can be translated to an expression in L_1 that describes the same set of variants.
L_1 is variant equivalent to L_2	**iff**	L_1 is as expressive as L_2 and vice versa.

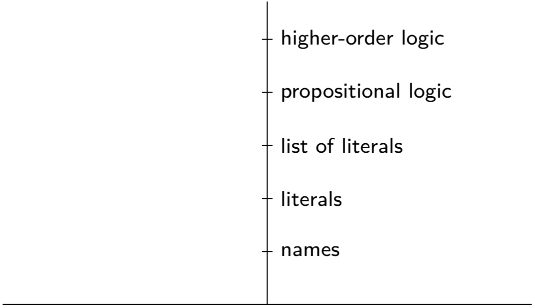
L_1 is as expressive as L_2	**iff**	Every expression in L_2 can be translated to an expression in L_1 that describes the same set of variants.
L_1 is variant equivalent to L_2	**iff**	L_1 is as expressive as L_2 and vice versa.
L_1 is semantically equivalent to L_2	**iff**	L_1 and L_2 are variant equivalent and same configurations yield same variants. (Translation of configurations is an isomorphism.)

Annotation Language
How to annotate elements
with variability information?

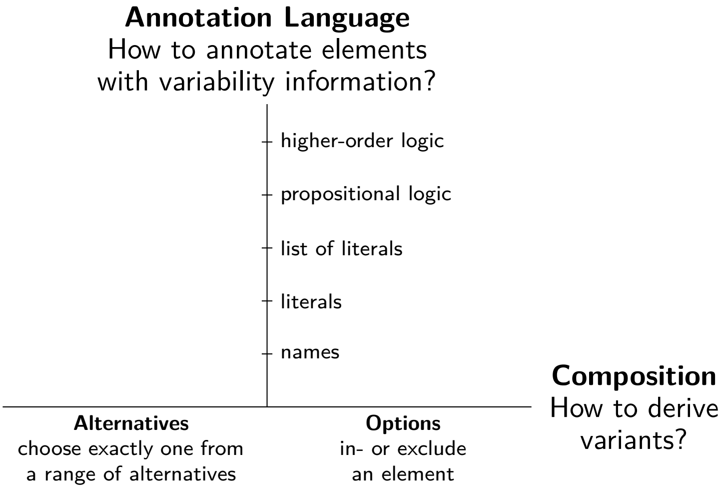


Composition
How to derive
variants?

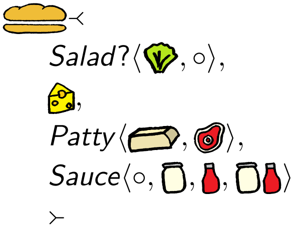
Annotation Language
How to annotate elements
with variability information?



Composition
How to derive
variants?



Annotation Language
How to annotate elements
with variability information?
higher-order logic
propositional logic
list of literals
literals
names
Composition
How to derive
variants?
Alternatives
choose exactly one from
a range of alternatives
Options
in- or exclude
an element



Annotation Language

How to annotate elements with variability information?

- higher-order logic
- propositional logic
- list of literals
- literals
- names

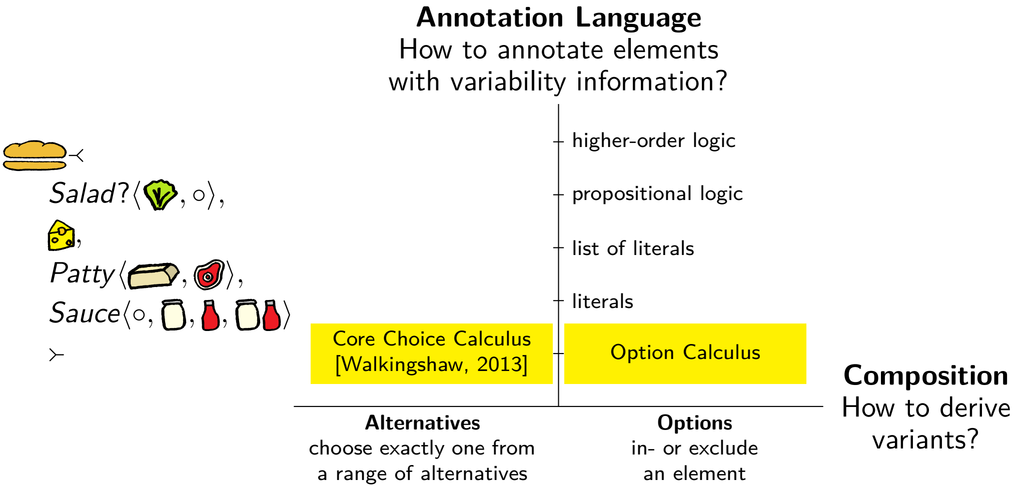
Core Choice Calculus [Walkingshaw, 2013]

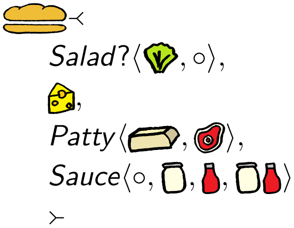
Alternatives
choose exactly one from a range of alternatives

Options
in- or exclude an element

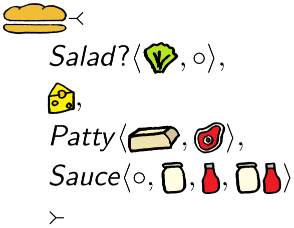
Composition

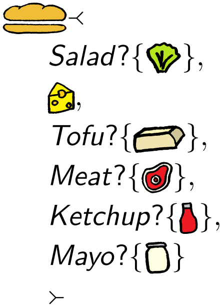
How to derive variants?


Annotation Language
How to annotate elements
with variability information?
Salad?⟨ , ○⟩,
,
Patty⟨ , ⟩,
Sauce⟨○, , , ⟩
higher-order logic
propositional logic
list of literals
literals
Core Choice Calculus
[Walkingshaw, 2013]
Option Calculus
Composition
How to derive
variants?
Alternatives
choose exactly one from
a range of alternatives
Options
in- or exclude
an element



Core Choice Calculus
[Walkingshaw, 2013]

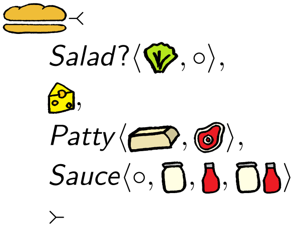




Core Choice Calculus [Walkingshaw, 2013]

Option Calculus

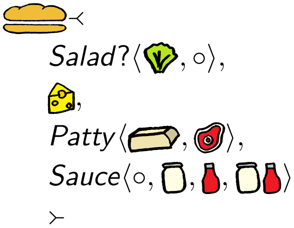
Named options cannot express alternatives!

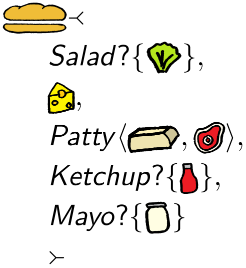


Core Choice Calculus [Walkingshaw, 2013]

Option Calculus

Named options cannot express alternatives!





Core Choice Calculus [Walkingshaw, 2013]

Option Calculus

Variation Trees [Bittner et al., 2022]

L is complete **iff** L can encode any given set of variants.

L is complete	**iff**	L can encode any given set of variants.
L_1 is complete $\wedge$ L_2 is as expressive as L_1	$\models$	L_2 is complete.

L is complete	**iff**	L can encode any given set of variants.
L_1 is complete $\wedge$ L_2 is as expressive as L_1	$\models$	L_2 is complete.
L_1 is complete $\wedge$ L_2 is incomplete	$\models$	L_2 is less expressive than L_1.


1 P 229600 HI 229600 PUSH START
KO
KEN
99
RYU
FIGHT!
Alternatives
Options
BATTLE 04

- complete

Alternatives

Options

- incomplete

- complete
- more expressive

Alternatives

Options

- incomplete
- less expressive

- complete
- more expressive
- encoding options requires neutral domain elements or sacrifices sharing

Alternatives

Options

- incomplete
- less expressive
- cannot encode alternatives

- complete
- more expressive
- encoding options requires neutral domain elements or sacrifices sharing

Alternatives

Options

- incomplete
- less expressive
- cannot encode alternatives

Conclusions:

- Options are useful syntax to increase sharing.
- For completeness, **else** statements or **negations** of annotations are essential.

Core
Choice Calculus

Binary
Choice Calculus

Option Calculus

Artifact Trees

Algebraic
Decision Diagrams

Variation Trees

Variability-Aware
Abstract Syntax Trees

Binary
Decision Diagrams

Contributions

Core
Choice Calculus

Binary
Choice Calculus

Option Calculus

Artifact Trees

Algebraic
Decision Diagrams

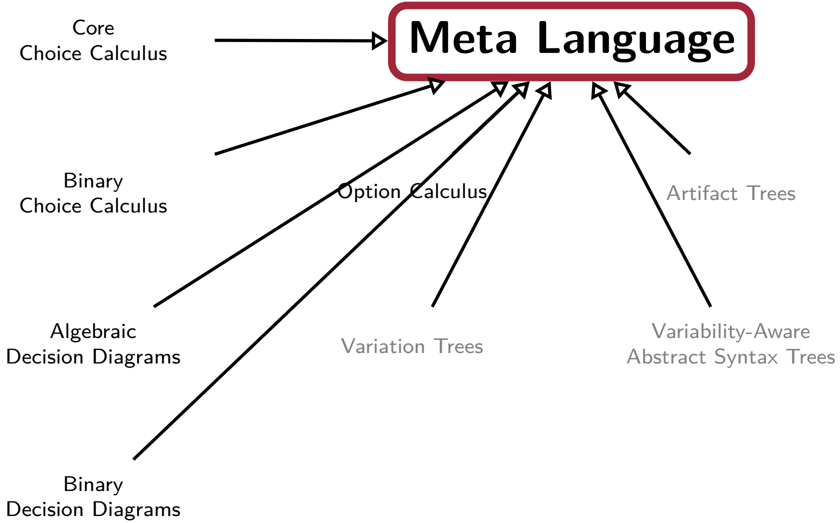
Variation Trees

Variability-Aware
Abstract Syntax Trees

Binary
Decision Diagrams

Contributions

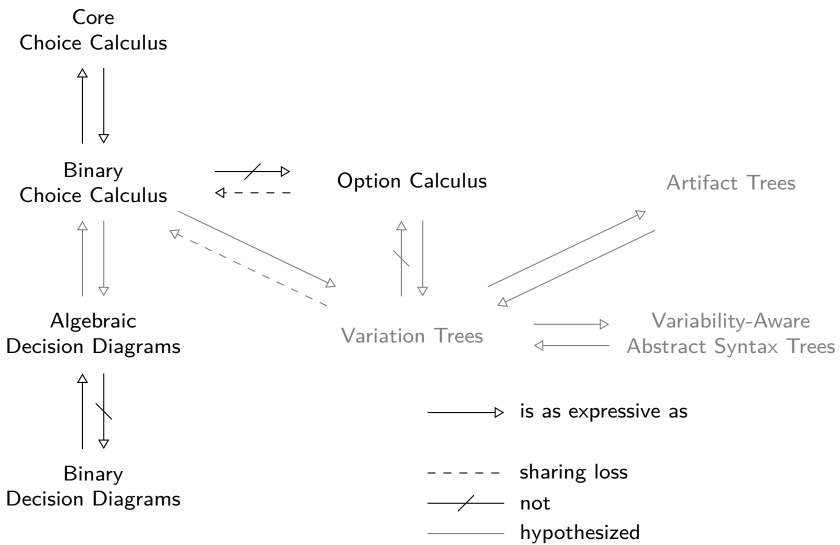
option calculus to formally clarify relationship between alternatives and options



Contributions

option calculus to formally clarify relationship between alternatives and options

formal framework based on meta-language for variability

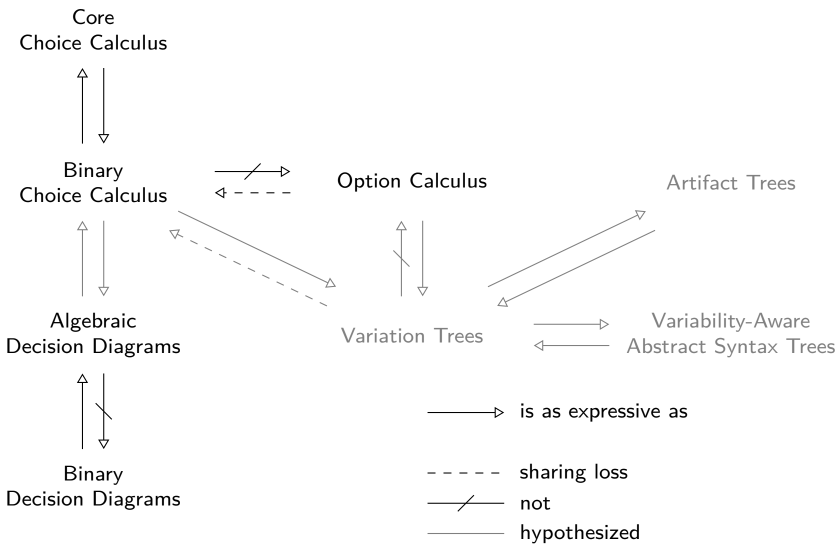


Contributions

option calculus to formally clarify relationship between alternatives and options

formal framework based on meta-language for variability

formal comparison of variability languages



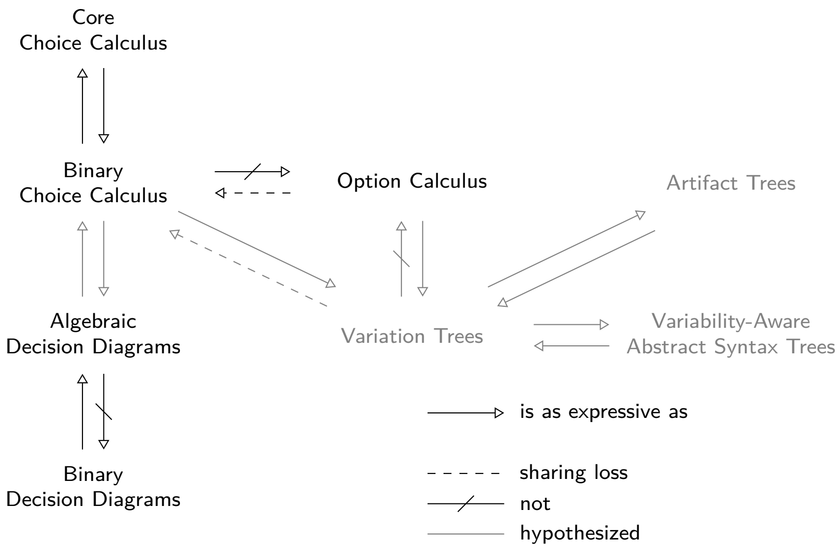
Contributions

option calculus to formally clarify relationship between alternatives and options

formal framework based on meta-language for variability

formal comparison of variability languages

(in)completeness proofs



Contributions

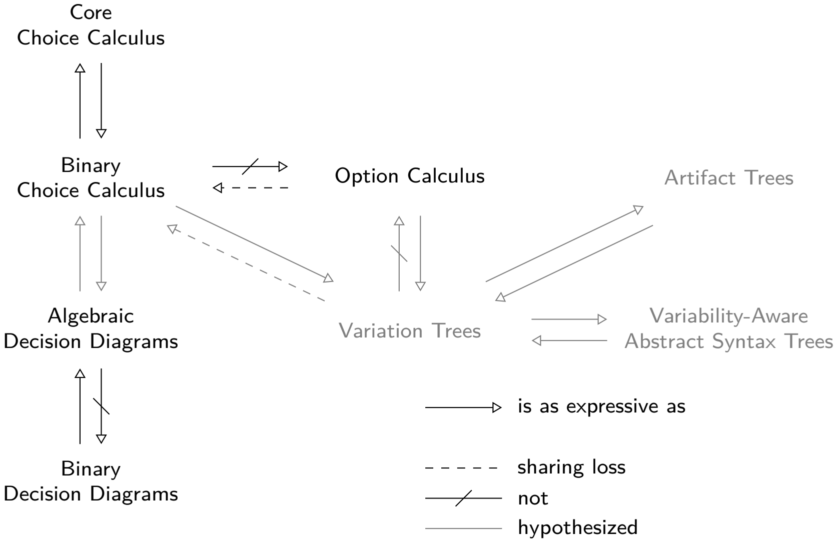
option calculus to formally clarify relationship between alternatives and options

formal framework based on meta-language for variability

formal comparison of variability languages

(in)completeness proofs

open-source Agda library



Contributions

option calculus to formally clarify relationship between alternatives and options

formal framework based on meta-language for variability

formal comparison of variability languages

(in)completeness proofs

open-source Agda library

but still WIP

Aadithya, K. V., Michalak, T. P., and Jennings, N. R. (2011).
Representation of Coalitional Games with Algebraic Decision Diagrams.
In *Proc. Int'l Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1121–1122. International Foundation for Autonomous Agents and Multiagent Systems.

Ataei, P., Khan, F., and Walkingshaw, E. (2021).
A Variational Database Management System.
In *Proc. Int'l Conf. on Generative Programming and Component Engineering (GPCE)*, pages 29–42. ACM.

Bahar, R. I., Frohm, E. A., Gaona, C. M., Hachtel, G. D., Macii, E., Pardo, A., and Somenzi, F. (1993).
Algebraic Decision Diagrams and Their Applications.
In *Proc. Int'l Conf. on Computer-Aided Design (ICCAD)*, pages 188–191. IEEE.

Bittner, P. M., Schultheiß, A., Thüm, T., Kehrer, T., Young, J. M., and Linsbauer, L. (2021).
Feature Trace Recording.
In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 1007–1020. ACM.

Bittner, P. M., Tinnes, C., Schultheiß, A., Viegener, S., Kehrer, T., and Thüm, T. (2022).
Classifying Edits to Variability in Source Code.
In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 196–208. ACM.

Castro, T., Teixeira, L., Alves, V., Apel, S., Cordy, M., and Gheyi, R. (2021).
A Formal Framework of Software Product Line Analyses.
Trans. on Software Engineering and Methodology (TOSEM), 30(3).

Chen, S., Erwig, M., and Walkingshaw, E. (2014).
Extending Type Inference to Variational Programs.
ACM Trans. on Programming Languages and Systems (TOPLAS), 36(1):1:1–1:54.

Kästner, C. (2010).
Virtual Separation of Concerns: Toward Preprocessors 2.0.
PhD thesis, University of Magdeburg.

Kästner, C., Giarrusso, P. G., Rendel, T., Erdweg, S., Ostermann, K., and Berger, T. (2011).
Variability-Aware Parsing in the Presence of Lexical Macros and Conditional Compilation.
In *Proc. Conf. on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 805–824. ACM.

Stănciulescu, S., Berger, T., Walkingshaw, E., and Wąsowski, A. (2016).
Concepts, Operations, and Feasibility of a Projection-Based Variation Control System.
In *Proc. Int'l Conf. on Software Maintenance and Evolution (ICSME)*, pages 323–333. IEEE.

Walkingshaw, E. (2013).
The Choice Calculus: A Formal Language of Variation.
PhD thesis, Oregon State University.

Walkingshaw, E. and Ostermann, K. (2014).
Projectional Editing of Variational Software.
In *Proc. Int'l Conf. on Generative Programming: Concepts & Experiences (GPCE)*, pages 29–38. ACM.

Young, J. M., Bittner, P. M., Walkingshaw, E., and Thüm, T. (2022).
Variational Satisfiability Solving: Efficiently Solving Lots of Related SAT Problems.

Empirical Software Engineering (EMSE), 28.