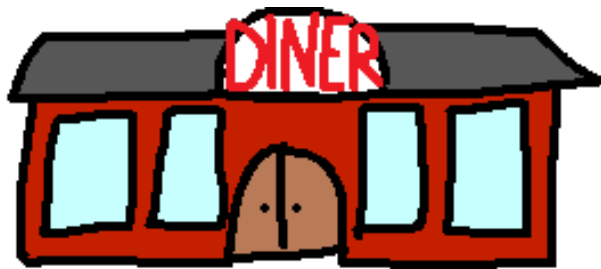


```
30 Since L is complete, there exists an expression
31 By assumption, M is as expressive as L.
32 Thus, there exists an expression  $m \in M$  that al
33 Since V was picked arbitrarily, M can encode a
34 Thus, M is complete.
35
36 Dual theorem to: soundness-by-expressiveness.
37 -}
38 completeness-by-expressiveness :  $\forall \{L\ M : \text{VariabilityLanguage}\ V\}$ 
39    $\rightarrow \text{Complete}\ M$ 
40    $\rightarrow L \geq M$ 
41   -----
42    $\rightarrow \text{Complete}\ L$ 
```

On the Expressive Power of Languages for Static Variability

Paul Bittner, A. Schultheiß, B. Moosherr, Y. Young, L. Teixeira, E. Walkingshaw, P. Ataei, T. Thüm | Oct 25 | OOPSLA





VEGETARIAN

WHICH WICH WOULD YOU LIKE?

☐ TRIPLE CHEESE MELT
☐ ELVIS WICH (P.S. Honey & Barbecue)
☐ 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?

☐ Yellow ☐ Dijon ☐ Honey ☐ Deli

☐ Regular ☐ Lite ☐ Horseradish ☐ Spicy

☐ BBQ ☐ Buffalo ☐ Marinara
☐ Thousand Island ☐ Ranch

☐ Red ☐ Grilled ☐ Crispy Strings

☐ Lettuce ☐ Tomato ☐ Pickles ☐ Jalapenos
☐ Olive Salad ☐ Mushrooms ☐ Sauerkraut
☐ Coleslaw ☐ Bell Peppers

☐ Oil ☐ Vinegar
☐ Onion ☐ Oregano ☐ Parmesan

☐ 75¢ Each ☐ Whole

This eerily
reminds me of
work...




Indeed! The sandwich variety resembles the variability in our code base!

This eerily reminds me of work...



Index
sandw
reser
variab
cod

```
static void  
f_foreground(/* params */)   
{  
#ifdef FEAT_GUI  
if (gui.in_use)  
gui_mch_set_foreground();  
#else  
#ifdef MSWIN  
win32_set_foreground();  
#endif  
#endif  
}
```

Vim Commit [ab4cece](#)

Index
sandw
reser
variab
cod

```
static void  
f_foreground(/* params */) {  
#ifdef FEAT_GUI  
    if (gui.in_use)  
        gui_mch_set_foreground();  
#else  
# ifdef MSWIN  
    win32_set_foreground();  
# endif  
#endif  
}
```

#define FEAT_GUI 1

```
static void  
f_foreground(/* params */) {  
    if (gui.in_use)  
        gui_mch_set_foreground();  
}
```

Vim Commit [ab4cece](#)

Index
sandw
reser
variab
cod

```
static void  
f_foreground(/* params */) {  
#ifdef FEAT_GUI  
    if (gui.in_use)  
        gui_mch_set_foreground();  
#else  
#ifdef MSWIN  
    win32_set_foreground();  
#endif  
#endif  
}
```

#define FEAT_GUI 1

#define FEAT_GUI 0,
#define MSWIN 1

```
static void  
f_foreground(/* params */) {  
    if (gui.in_use)  
        gui_mch_set_foreground();  
}
```

```
static void  
f_foreground(/* params */) {  
    win32_set_foreground();  
}
```

Vim Commit [ab4cece](#)

Index
sandw
reser
variab
cod

```
static void  
f_foreground(/* params */) {  
#ifdef FEAT_GUI  
    if (gui.in_use)  
        gui_mch_set_foreground();  
#else  
#ifdef MSWIN  
    win32_set_foreground();  
#endif  
#endif  
}
```

#define FEAT_GUI 1

#define FEAT_GUI 0,
#define MSWIN 1

#define FEAT_GUI 0,
#define MSWIN 0

```
static void  
f_foreground(/* params */) {  
    if (gui.in_use)  
        gui_mch_set_foreground();  
}
```

```
static void  
f_foreground(/* params */) {  
    win32_set_foreground();  
}
```

```
static void  
f_foreground(/* params */) {  
  
}
```

Vim Commit [ab4cece](#)

And its not just
our code and
sandwiches, I
guess...



VEGETARIAN

WHICH WICH WOULD YOU LIKE?

☐ TRIPLE CHEESE MELT
☐ ELVIS WICH (P.S. Honey & Barbecue)
☐ 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
☐ Thousand Island ☐ Ranch

ONIONS
☐ Red ☐ Grilled ☐ Crispy Strings

VEGGIES
☐ Lettuce ☐ Tomato ☐ Pickles ☐ Jalapenos
☐ Olive Salad ☐ Mushrooms ☐ Sauerkraut
☐ Coleslaw ☐ Bell Peppers

OILS & SPICES
☐ Oil ☐ Vinegar
☐ Mustard ☐ Oregano ☐ Parmesan
☐ Soy Sauce ☐ Teriyaki ☐ Worcestershire

True, there is
variability in many
domains!



True, there is
variability in many
domains!



Customise your wheel

Finish

15" black alloy wheels (5-
double-spoke)



Cap colour

Black small centre cap



True, there is variability in many domains!



Customise your wheel

Finish

15" black alloy wheels (5-double-spoke)

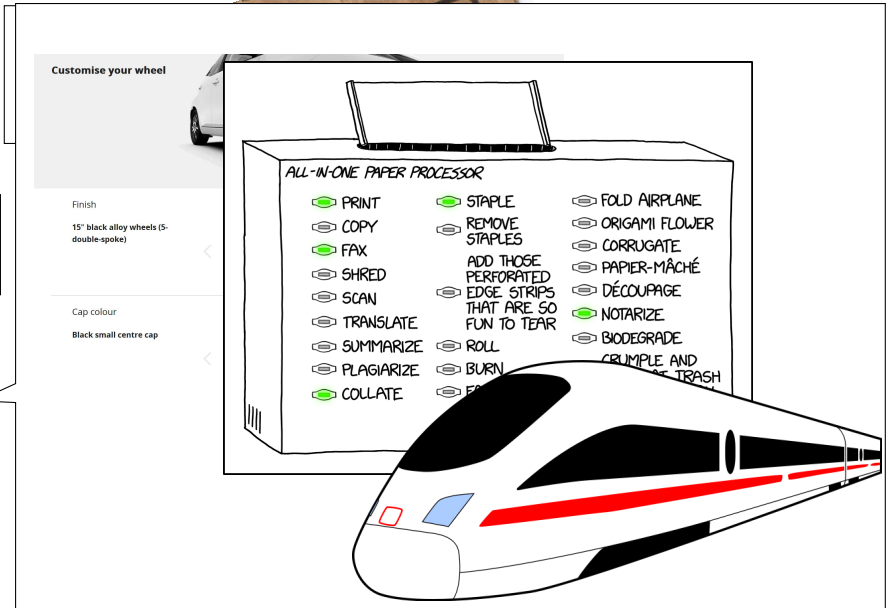
Cap colour

Black small centre cap

ALL-IN-ONE PAPER PROCESSOR

<input checked="" type="checkbox"/> PRINT	<input checked="" type="checkbox"/> STAPLE	<input type="checkbox"/> FOLD AIRPLANE
<input type="checkbox"/> COPY	<input type="checkbox"/> REMOVE STAPLES	<input type="checkbox"/> ORIGAMI FLOWER
<input checked="" type="checkbox"/> FAX	ADD THOSE PERFORATED	<input type="checkbox"/> CORRUGATE
<input type="checkbox"/> SHRED	EDGE STRIPS THAT ARE SO FUN TO TEAR	<input type="checkbox"/> PAPIER-MÂCHÉ
<input type="checkbox"/> SCAN	<input type="checkbox"/> ROLL	<input type="checkbox"/> DÉCOUPAGE
<input type="checkbox"/> TRANSLATE	<input type="checkbox"/> BURN	<input checked="" type="checkbox"/> NOTARIZE
<input type="checkbox"/> SUMMARIZE	<input type="checkbox"/> EAT	<input type="checkbox"/> BIODEGRADE
<input type="checkbox"/> PLAGIARIZE		<input checked="" type="checkbox"/> CRUMPLE AND THROW AT TRASH LIKE BASKETBALL
<input checked="" type="checkbox"/> COLLABORATE		

True, there is variability in many domains!



True, there is variability in many domains!



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
☒ COPY ☒ REMOVE ☒ ORIGAMI FLOWER

Microsoft Productivity Software

Microsoft Office 365 Home **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



But how to
describe and analyze
variability across domains?





	Computation	Static Variability
Implementation	Java, C, Haskell, OCaml, Prolog, ...	
Specification		

	Computation	Static Variability
Implementation	Java, C, Haskell, OCaml, Prolog, ...	
Specification	Turing Machine, Lambda Calculus, Type-0 Grammar, ...	

	Computation	Static Variability
Implementation	Java, C, Haskell, OCaml, Prolog, ...	C Preprocessor, Mixins, Aspects, Sandwich Menus, ...
Specification	Turing Machine, Lambda Calculus, Type-0 Grammar, ...	

	Computation	Static Variability
Implementation	Java, C, Haskell, OCaml, Prolog, ...	C Preprocessor, Mixins, Aspects, Sandwich Menus, ...
Specification	Turing Machine, Lambda Calculus, Type-0 Grammar, ...	?


Core Choice Calculus ^[Erwig and Walkingshaw, 2011] – A Lambda Calculus of Variability? _[Walkingshaw, 2013]

$$\begin{array}{lcl} e & ::= & a\langle e, \dots, e \rangle \quad \textit{Object Structure} \\ & | & D\langle e, \dots, e \rangle \quad \textit{Choice} \end{array}$$

Core Choice Calculus ^[Erwig and Walkingshaw, 2011] – A Lambda Calculus of Variability? _[Walkingshaw, 2013]



$$e ::= a\langle e, \dots, e \rangle \quad \text{Object Structure}$$
$$| D\langle e, \dots, e \rangle \quad \text{Choice}$$

always 

maybe 

always 

either  or 

any combination of  and 


always 

Core Choice Calculus ^[Erwig and Walkingshaw, 2011] – A Lambda Calculus of Variability? _[Walkingshaw, 2013]

$$e ::= a \langle e, \dots, e \rangle \quad \text{Object Structure}$$



$$| D \langle e, \dots, e \rangle \quad \text{Choice}$$

always 










maybe 

always 

either  or 

any combination of  and 

always 


 \langle
Salad \langle  \rangle ,
,
Patty \langle   \rangle ,
Sauce \langle     \rangle
 \rangle

Core Choice Calculus ^[Erwig and Walkingshaw, 2011] – A Lambda Calculus of Variability? _[Walkingshaw, 2013]

$$e ::= a\langle e, \dots, e \rangle \quad \text{Object Structure}$$



$$| D\langle e, \dots, e \rangle \quad \text{Choice}$$

always 








maybe 






always 

either  or 

any combination of  and 

always 

 \langle
Salad \langle  \rangle ,
 \rangle ,
Patty \langle  \rangle ,
Sauce \langle \circ ,  \rangle ,  \rangle ,  \rangle
 $\rangle \rangle (c)$

 \langle  \rangle ,  \rangle ,  \rangle ,  \rangle
 $=$ if $c(\text{Salad}) = 0$,
 $c(\text{Patty}) = 0$,
 $c(\text{Sauce}) = 2$.

Core Choice Calculus [Erwig and Walkingshaw, 2011] – A Lambda Calculus of Variability?

always

maybe

always

either

any combination of

always



= 0,

= 0,

= 2.

Choice Calculus

[Erwig and Walkingshaw, 2011]

[Walkingshaw, 2013]

Artifact Trees

[Linsbauer et al., 2017]

Feature
Structure Trees

[Apel et al., 2013]

Algebraic
Decision Trees

[Bahar et al., 1993]

Gruler's Language

[Gruler, 2010]

Clone and Own

Variability-
Aware ASTs

[Kästner et al., 2008]

Variational IMP

[Midtgaard et al., 2015]

. . .

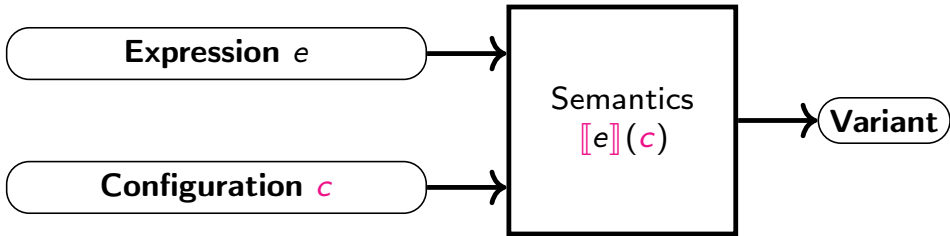
But how do these languages relate?
Can we transfer research results based
on one formalism to the others?

Variability-
Aware ASTs
[Kästner et al., 2008]

Variational IMP
[Midtgaard et al., 2010]

...

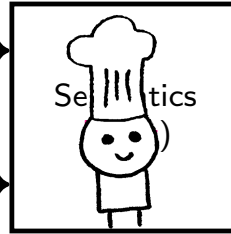






Expression e

Configuration c

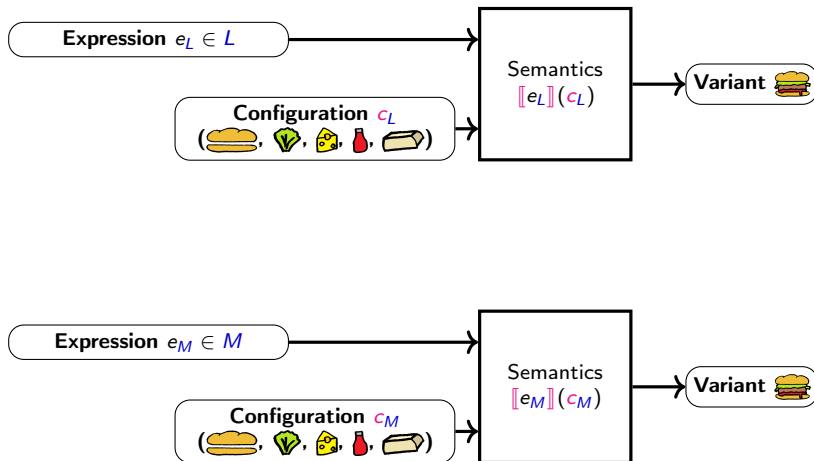


Variant

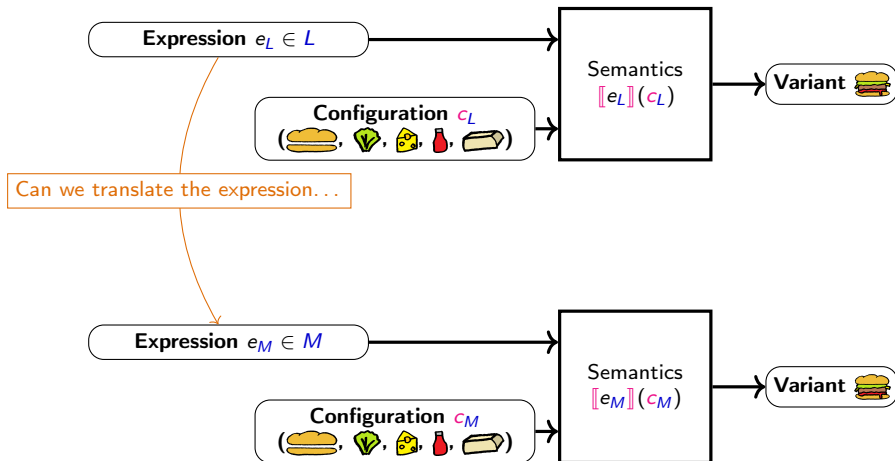


How to Compare Variability Languages L and M semantically?

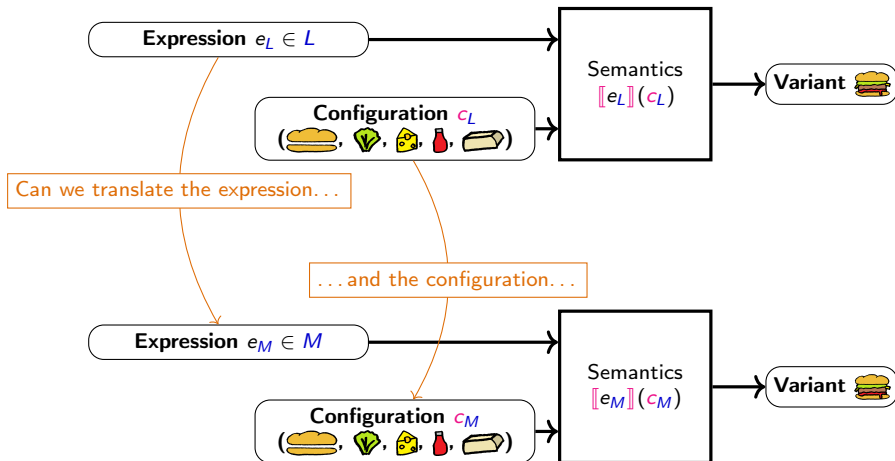
How to Compare Variability Languages L and M semantically?



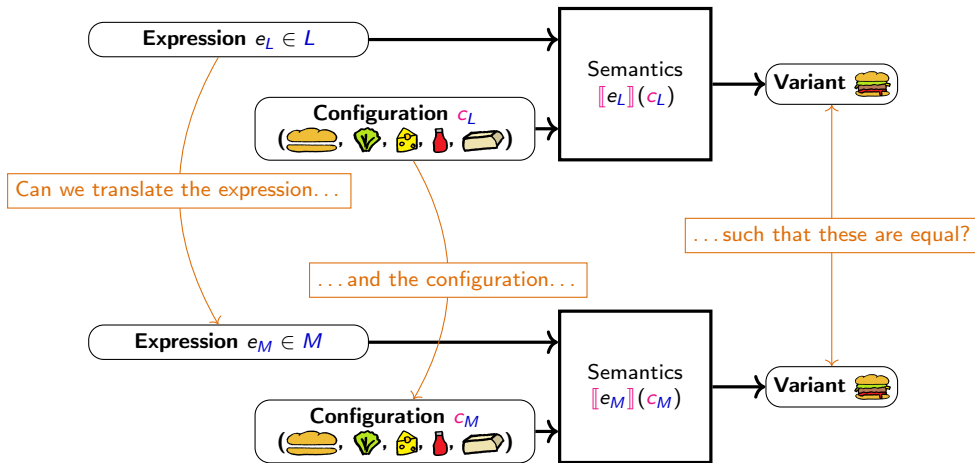
How to Compare Variability Languages L and M semantically?



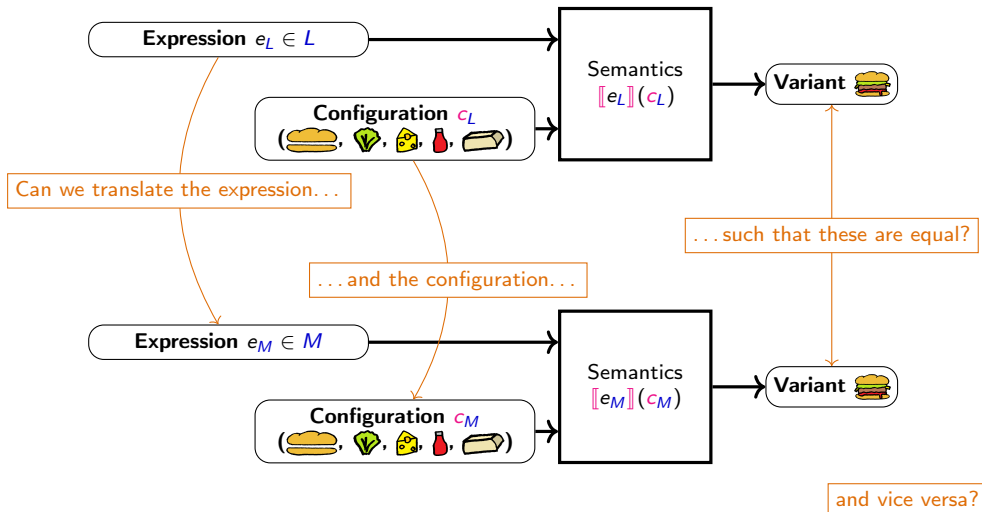
How to Compare Variability Languages L and M semantically?



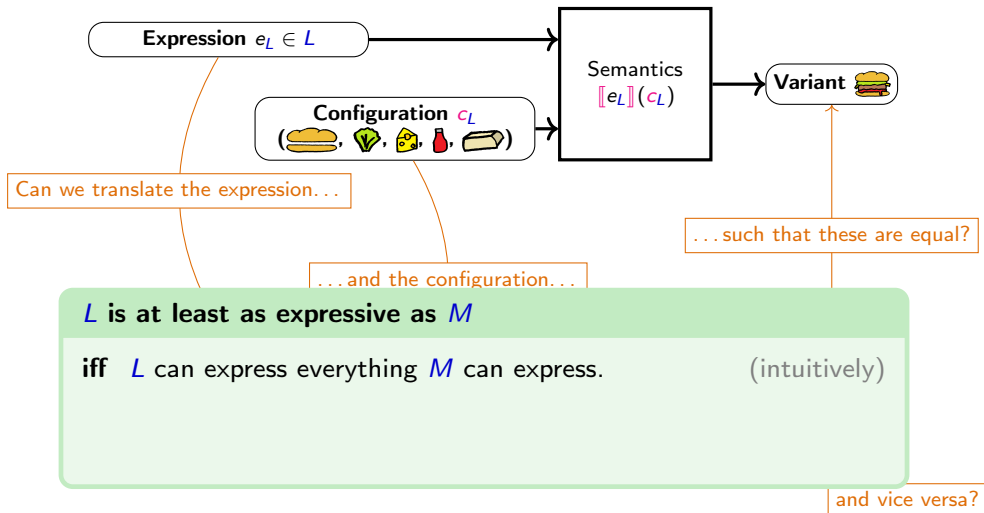
How to Compare Variability Languages L and M semantically?



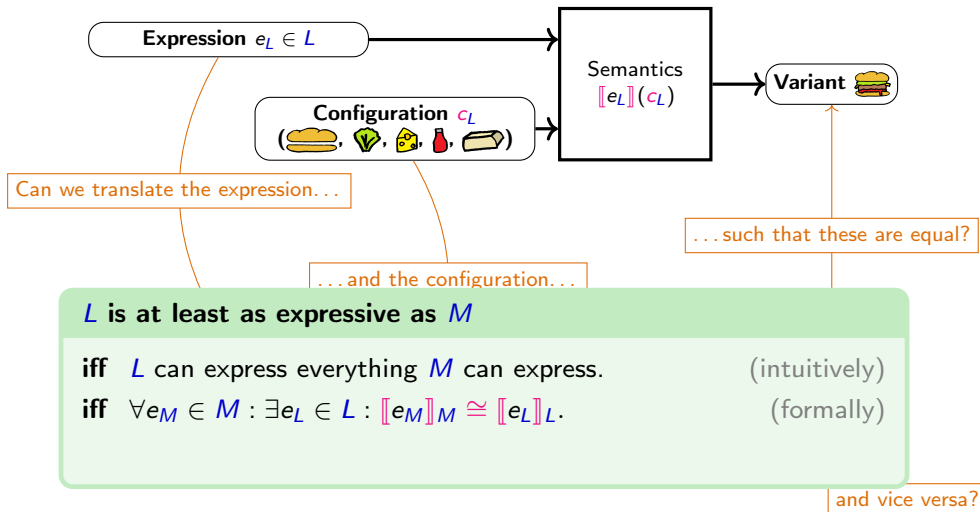
How to Compare Variability Languages L and M semantically?



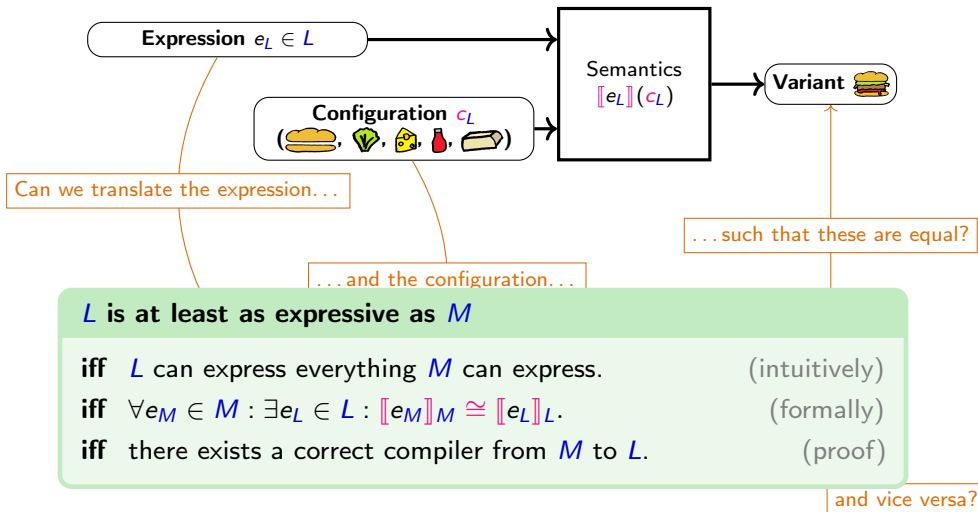
How to Compare Variability Languages L and M semantically?



How to Compare Variability Languages L and M semantically?



How to Compare Variability Languages L and M semantically?



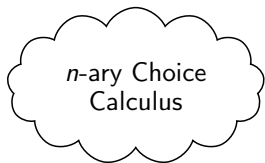
Binary Choice
Calculus

Option
Calculus

Algebraic
Decision Trees

Core Choice
Calculus

Feature
Structure Trees



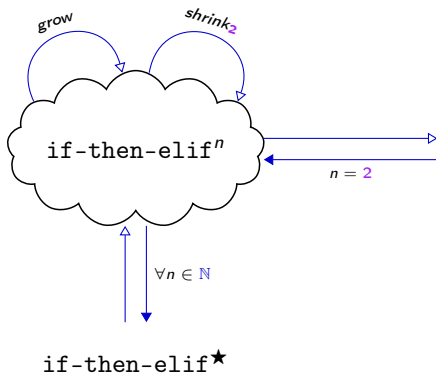
Binary Choice
Calculus

Option
Calculus

Algebraic
Decision Trees

Core Choice
Calculus

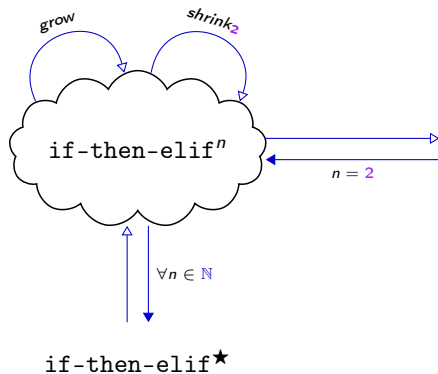
Feature
Structure Trees



Option
Calculus

Algebraic
Decision Trees

Feature
Structure Trees



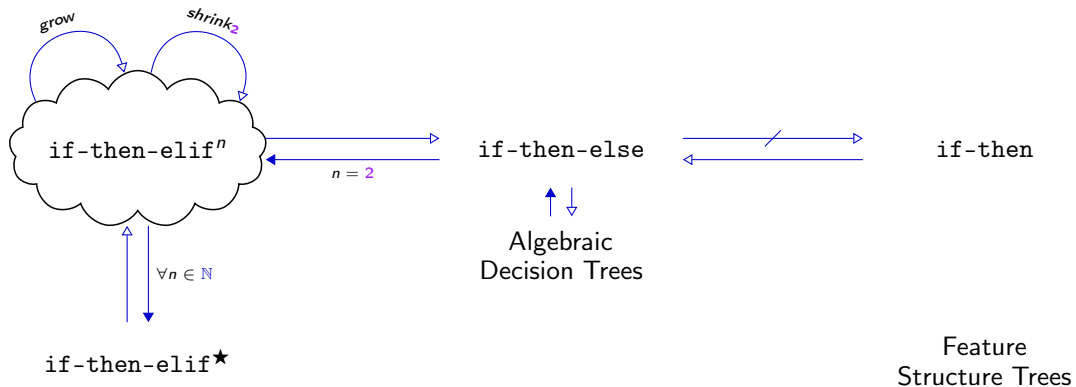
`if-then-else`

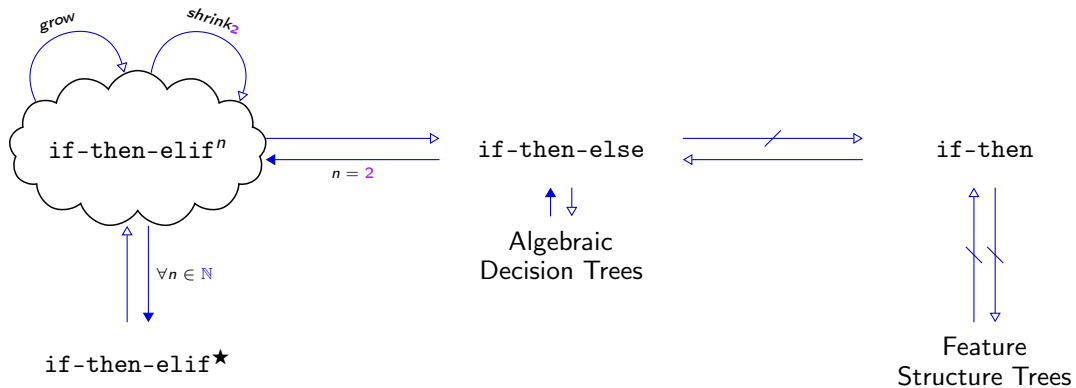


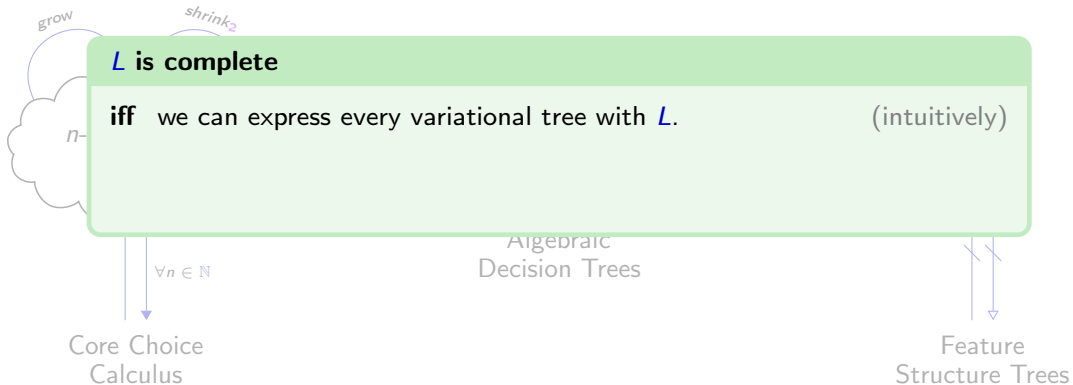
Algebraic
Decision Trees

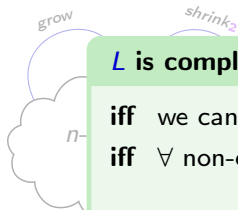
Option
Calculus

Feature
Structure Trees









L is complete

iff we can express every variational tree with L . (intuitively)

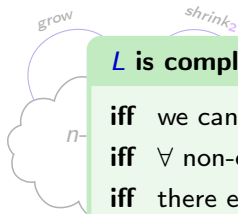
iff \forall non-empty, finite set of variants $V : \exists e \in L : \llbracket e \rrbracket \cong V$. (formally)

$\forall n \in \mathbb{N}$

Core Choice
Calculus

Algebraic
Decision Trees

Feature
Structure Trees



L is complete

iff we can express every variational tree with L . (intuitively)

iff \forall non-empty, finite set of variants $V : \exists e \in L : \llbracket e \rrbracket \cong V$. (formally)

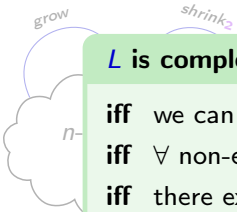
iff there exists a correct n -way *differencing* algorithm. (proof)

$\forall n \in \mathbb{N}$

Core Choice
Calculus

Algebraic
Decision Trees

Feature
Structure Trees



L is complete

iff we can express every variational tree with L . (intuitively)

iff \forall non-empty, finite set of variants $V : \exists e \in L : \llbracket e \rrbracket \cong V$. (formally)

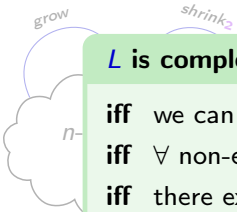
iff there exists a correct n -way *differencing* algorithm. (proof)

Algebraic

L is sound

iff we can express *only* variational trees with L . (intuitively)

rees



L is complete

iff we can express every variational tree with L . (intuitively)

iff \forall non-empty, finite set of variants $V : \exists e \in L : \llbracket e \rrbracket \cong V$. (formally)

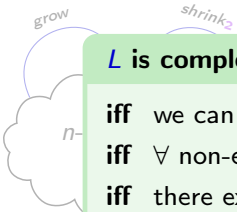
iff there exists a correct n -way *differencing* algorithm. (proof)

Algebraic

L is sound

iff we can express *only* variational trees with L . (intuitively)

iff $\forall e \in L : \exists$ non-empty, finite set of variants $V : \llbracket e \rrbracket \cong V$. (formally) trees



L is complete

iff we can express every variational tree with L . (intuitively)

iff \forall non-empty, finite set of variants $V : \exists e \in L : \llbracket e \rrbracket \cong V$. (formally)

iff there exists a correct n -way *differencing* algorithm. (proof)

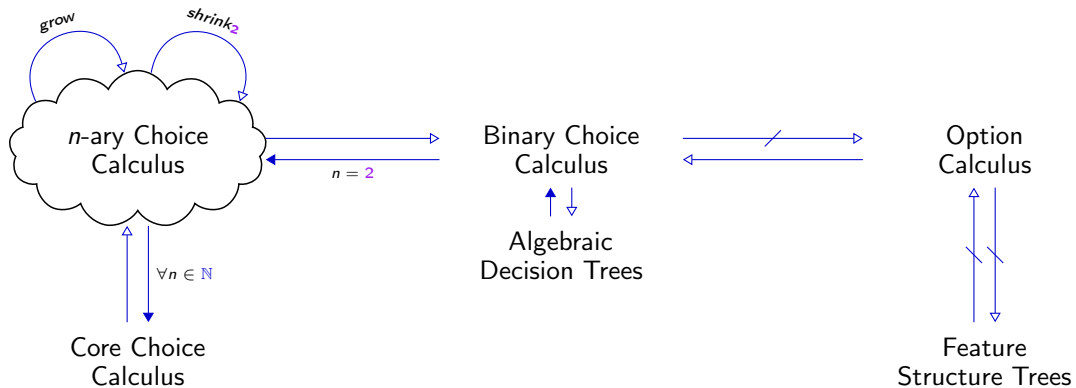
Algebraic

L is sound

iff we can express *only* variational trees with L . (intuitively)

iff $\forall e \in L : \exists$ non-empty, finite set of variants $V : \llbracket e \rrbracket \cong V$. (formally) trees

iff all configurations can be enumerated. (proof)



grow

shrink₂

GOURMET PIZZAS

	SM.	MED.	LG.	X-LG	JUMBO
Company Special	13.49	16.49	20.99	24.99	29.99
<i>Pepperoni, Ham, Fresh Mushrooms, Black Olives, Onions, Green Bell Peppers and Mozzarella Cheese</i>					
Pastrami Pizza	13.49	16.49	20.99	24.99	32.99
<i>Pastrami, Mustard, Pickles, and Mozzarella Cheese</i>					
"Big G"	12.99	15.99	20.99	23.99	28.99
<i>Creamy Alfredo Sauce, Imported Artichoke Hearts, Fresh Roasted Chicken Breast, Mozzarella and Pecorino Romano Cheese</i>					
Veggie Special	12.99	15.99	19.99	23.99	29.99
<i>Fresh Mushrooms, Black Olives, Onions, Green Bell Peppers, Fresh Tomatoes, Fresh Garlic, Spinach, Mozzarella, and Pecorino Romano Cheese</i>					
Teriyaki Chicken	12.99	15.99	19.99	23.99	29.99
<i>Fresh Roasted Chicken, Onions, Green Bell Peppers, Fresh Tomatoes, Pineapple, Mozzarella Cheese, Drizzled with Teriyaki Sauce</i>					
BBQ Chicken	12.99	15.99	19.99	23.99	29.99
<i>Fresh Roasted Chicken, Red Onions, Green Bell Peppers, Mozzarella Cheese, Drizzled with Smokey BBQ Sauce, Topped with fresh Cilantro</i>					
Margarita	11.99	14.99	18.99	22.99	28.99
<i>Fresh Basil, Fresh Roma Tomatoes, Mozzarella Cheese, Pecorino Romano Cheese, Drizzled with Imported EVOO</i>					
Meat Lover	13.49	16.49	20.99	24.99	29.99
<i>Pepperoni, Ham, Real Bacon, Italian Sausage and Mozzarella</i>					
Chicken Pesto	12.99	15.99	19.99	23.99	28.99
<i>Fresh Roasted Chicken, Fresh Basil Pesto (no pine nuts) Sun Dried Tomatoes and Mozzarella</i>					
Buffalo Chicken Ranch	13.49	16.49	20.99	24.99	29.99
<i>Spicy Buffalo Sauce, Mozzarella Cheese, Buffalo Seasoned Chicken, Buttermilk Ranch</i>					
White Pizza	12.99	15.99	19.99	23.99	29.99
<i>Ricotta Cheese, Mozzarella Cheese, Pecorino Romano Cheese, Oregano, Garlic, Olive Oil</i>					
Chicken Bacon Ranch	13.49	16.49	20.99	24.99	29.99
<i>Buttermilk Ranch, Mozzarella Cheese, Seasoned Chicken Breast, Real Bacon Bits</i>					

All pizzas served on a Hand-Tossed Thick Crust
Thin Crust and Gluten-Free options
available upon request

Binary Choice
CalculusAlgebraic
Decision Trees

Clone and Own

Option
CalculusFeature
Structure Trees

grow

shrink₂

GOURMET PIZZAS

	SM.	MED.	LG.	X-LG	JUMBO
Company Special	13.49	16.49	20.99	24.99	29.99
<i>Pepperoni, Ham, Fresh Mushrooms, Black Olives, Onions, Green Bell Peppers and Mozzarella Cheese</i>					
Pastrami Pizza	13.49	16.49	20.99	24.99	32.99
<i>Pastrami, Mustard, Pickles, and Mozzarella Cheese</i>					
"Big G"	12.99	15.99	20.99	23.99	28.99
<i>Creamy Alfredo Sauce, Imported Artichoke Hearts, Fresh Roasted Chicken Breast, Mozzarella and Pecorino Romano Cheese</i>					
Veggie Special	12.99	15.99	19.99	23.99	29.99
<i>Fresh Mushrooms, Black Olives, Onions, Green Bell Peppers, Fresh Tomatoes, Fresh Garlic, Spinach, Mozzarella, and Pecorino Romano Cheese</i>					
Teriyaki Chicken	12.99	15.99	19.99	23.99	29.99
<i>Fresh Roasted Chicken, Onions, Green Bell Peppers, Fresh Tomatoes, Pineapple, Mozzarella Cheese, Drizzled with Teriyaki Sauce</i>					
BBQ Chicken	12.99	15.99	19.99	23.99	29.99
<i>Fresh Roasted Chicken, Red Onions, Green Bell Peppers, Mozzarella Cheese, Drizzled with Smokey BBQ Sauce, Topped with fresh Cilantro</i>					
Margarita	11.99	14.99	18.99	22.99	28.99
<i>Fresh Basil, Fresh Roma Tomatoes, Mozzarella Cheese, Pecorino Romano Cheese, Drizzled with Imported EVOO</i>					
Meat Lover	13.49	16.49	20.99	24.99	29.99
<i>Pepperoni, Ham, Real Bacon, Italian Sausage and Mozzarella</i>					
Chicken Pesto	12.99	15.99	19.99	23.99	28.99
<i>Fresh Roasted Chicken, Fresh Basil Pesto (no pine nuts) Sun Dried Tomatoes and Mozzarella</i>					
Buffalo Chicken Ranch	13.49	16.49	20.99	24.99	29.99
<i>Spicy Buffalo Sauce, Mozzarella Cheese, Buffalo Seasoned Chicken, Buttermilk Ranch</i>					
White Pizza	12.99	15.99	19.99	23.99	29.99
<i>Ricotta Cheese, Mozzarella Cheese, Pecorino Romano Cheese, Oregano, Garlic, Olive Oil</i>					
Chicken Bacon Ranch	13.49	16.49	20.99	24.99	29.99
<i>Buttermilk Ranch, Mozzarella Cheese, Seasoned Chicken Breast, Real Bacon Bits</i>					

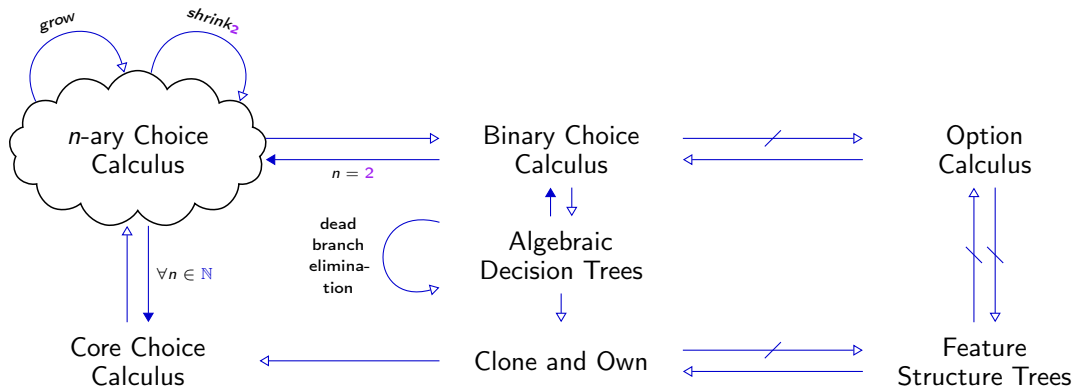
All pizzas served on a Hand-Tossed Thick Crust
Thin Crust and Gluten-Free options
available upon request

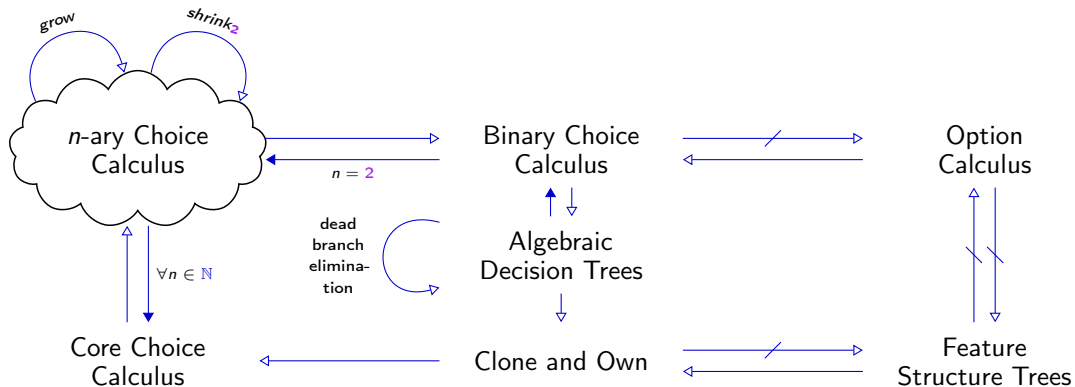
Binary Choice
CalculusAlgebraic
Decision Trees

Clone and Own

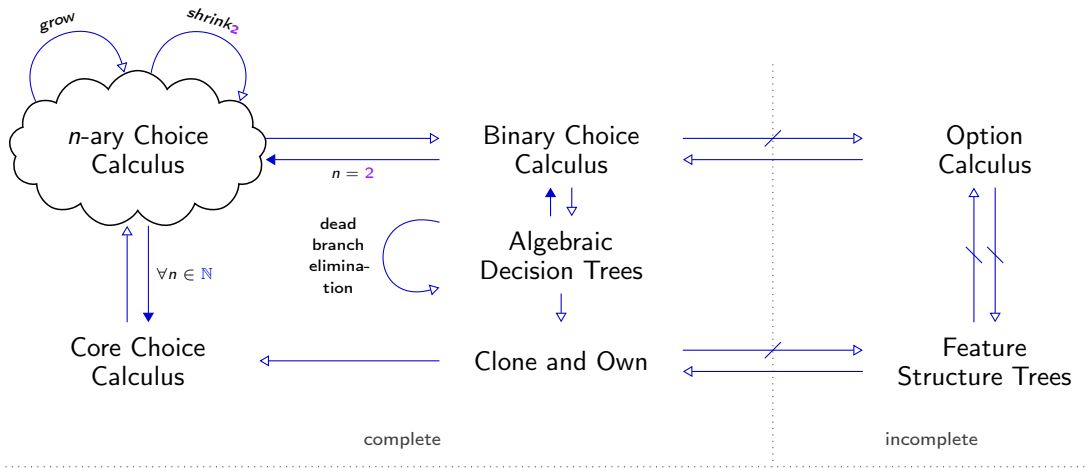
Option
CalculusFeature
Structure Trees

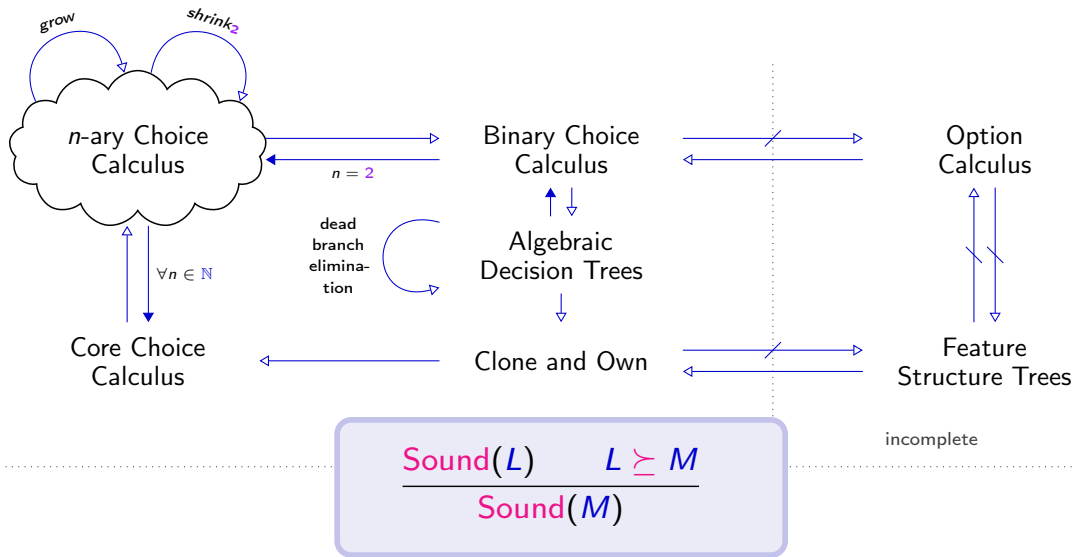
We prove this sound
and complete.

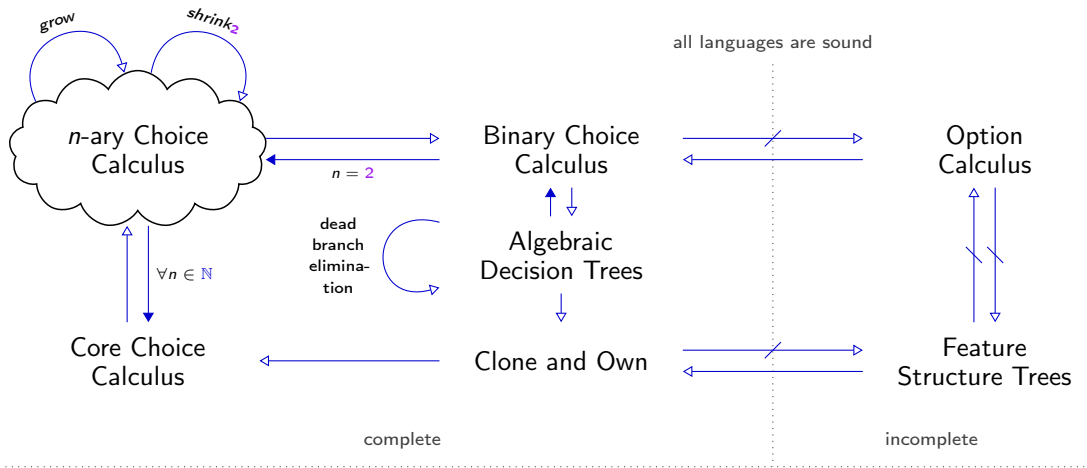


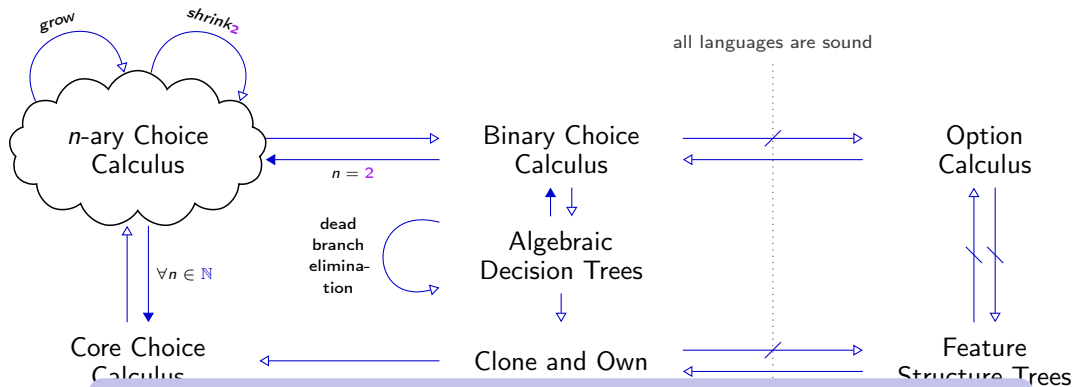


$$\frac{\text{Complete}(L) \quad M \succeq L}{\text{Complete}(M)}$$



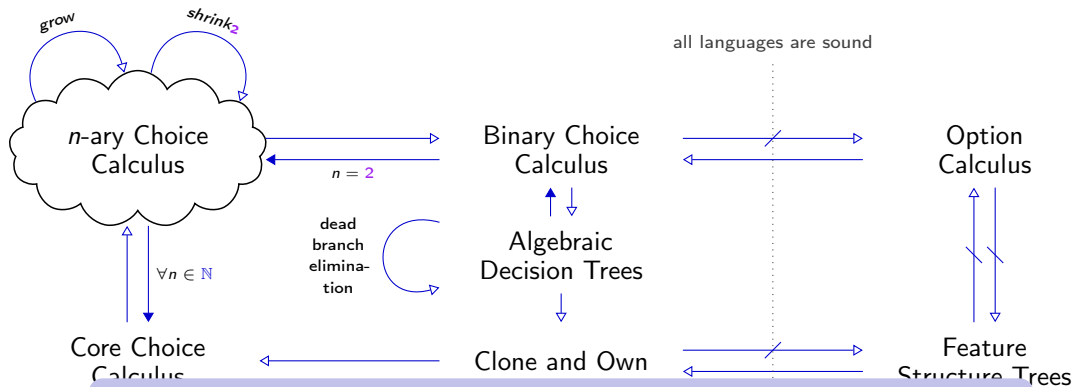






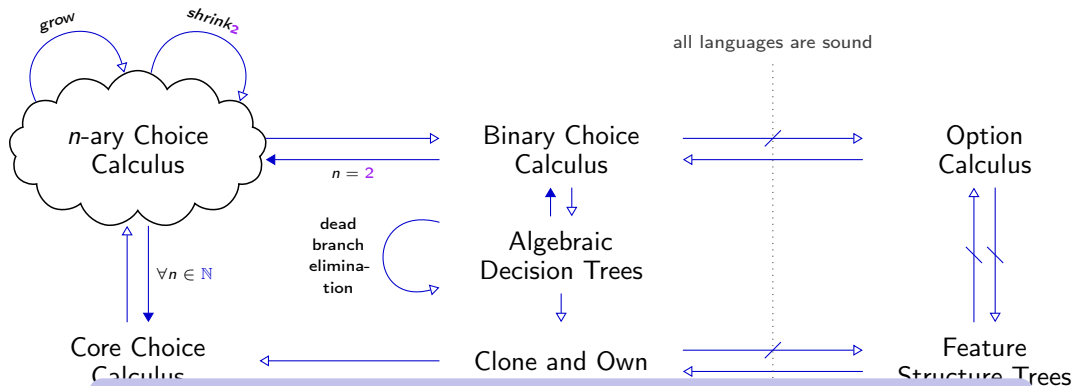
Conclusions

- Choice calculus is indeed a lambda calculus of variation



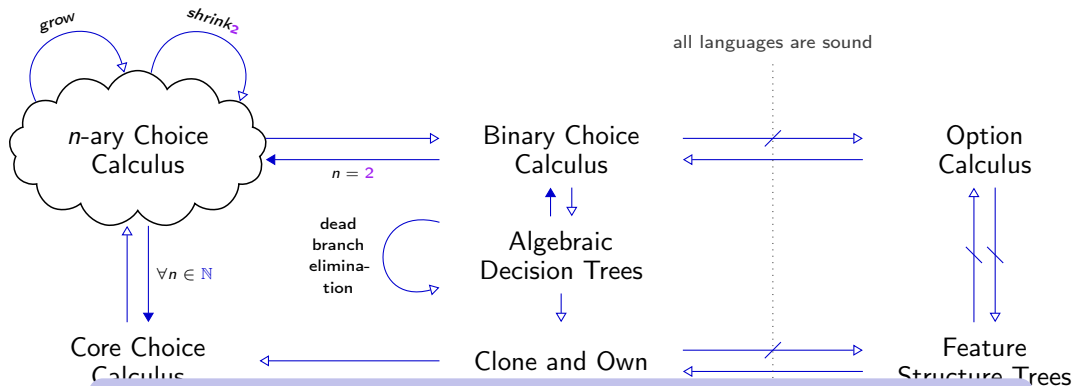
Conclusions

- Choice calculus is indeed a lambda calculus of variation as well as algebraic decision trees and clone-and-own.



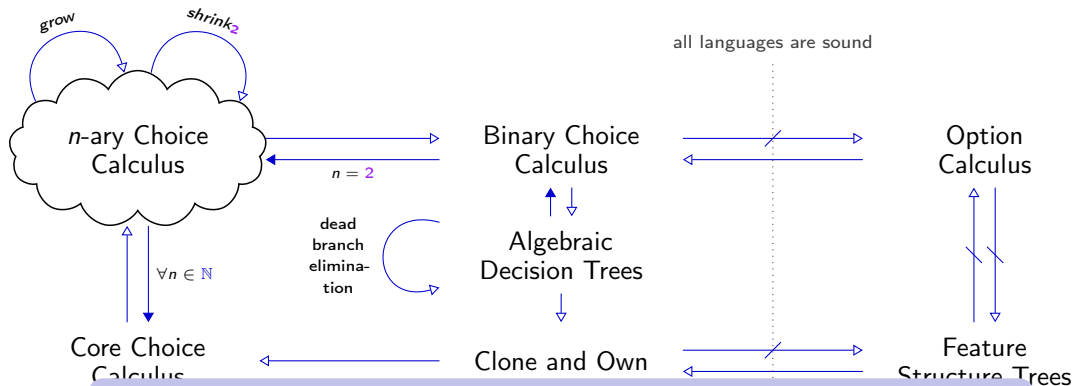
Conclusions

- Choice calculus is indeed a lambda calculus of variation as well as algebraic decision trees and clone-and-own.
- There are ≥ 3 classes of expressiveness, arising from different syntactical restrictions.



Conclusions

- Choice calculus is indeed a lambda calculus of variation as well as algebraic decision trees and clone-and-own. \Rightarrow **Are all languages equally good?**
- There are ≥ 3 classes of expressiveness, arising from different syntactical restrictions.



Conclusions

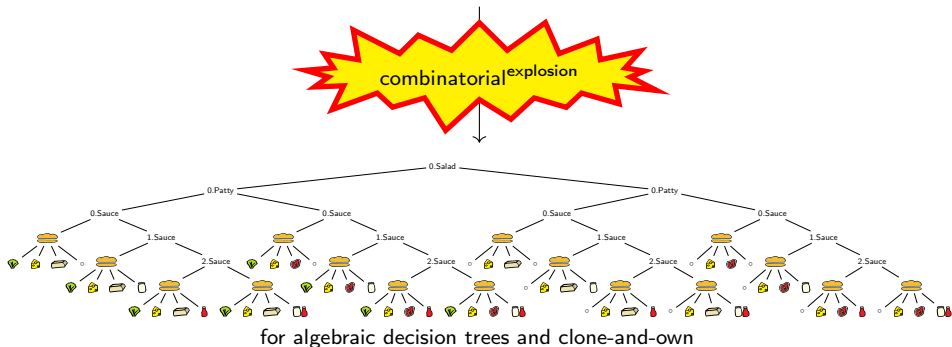
- Choice calculus is indeed a lambda calculus of variation as well as algebraic decision trees and clone-and-own. \Rightarrow **Are all languages equally good?**
- There are ≥ 3 classes of expressiveness, arising from different syntactical restrictions. \Rightarrow **What is the value of incomplete languages?**

Are all complete languages equally useful?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{🧀}, \text{Patty} \langle \text{🍔}, \text{🍖} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$

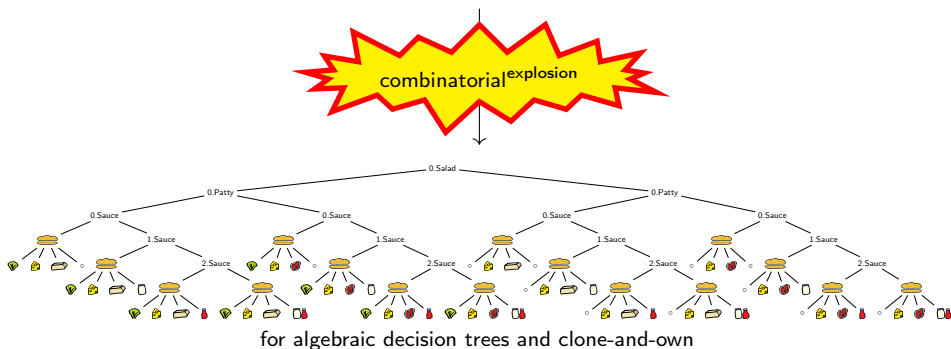
Are all complete languages equally useful?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{Patty} \langle \text{🍔}, \text{🍖} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$



Are all complete languages equally useful?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{Patty} \langle \text{🍔}, \text{🍖} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$



No, avoiding duplication of shared sub-terms is essential for scalability and usability.
(known problem of clone-and-own and a basic principle of software engineering)

What is the value of incomplete languages?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{🧀}, \text{Patty} \langle \text{🍔}, \text{🍖} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$

What is the value of incomplete languages?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{🧀}, \text{Patty} \langle \text{🍔}, \text{🍷} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$

Encoding options with choices requires context-sensitive neutral values \circ

What is the value of incomplete languages?

 $\langle \text{Salad} \langle \text{ , \circ \rangle , \text{ } , \text{Patty} \langle \text{ } , \text{ } \rangle , \text{Sauce} \langle \circ , \text{ } , \text{ } , \text{ } , \text{ } \rangle \rangle$

Encoding options with choices requires context-sensitive neutral values \circ or sacrifices on sharing:

$\text{Salad} \langle \text{ } \langle \text{ } , \text{ } , \text{Patty} \langle \text{ } , \text{ } \rangle , \text{Sauce} \langle \circ , \text{ } , \text{ } , \text{ } , \text{ } \rangle \rangle ,$
 $\text{ } \langle \text{ } , \text{Patty} \langle \text{ } , \text{ } \rangle , \text{Sauce} \langle \circ , \text{ } , \text{ } , \text{ } , \text{ } \rangle \rangle$

What is the value of incomplete languages?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{🧀}, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$

Encoding options with choices requires context-sensitive neutral values \circ or sacrifices on sharing:

$\text{Salad} \langle \text{🍔} \langle \text{🥬}, \text{🧀}, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle, \text{🍔} \langle \text{🥬}, \text{🧀}, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle \rangle$

 $\langle \text{Salad} \langle \text{🥬}, \text{🧀}, \text{Tofu} \langle \text{🍔} \rangle, \text{Meat} \langle \text{🥩} \rangle, \text{Ketchup} \langle \text{🍷} \rangle, \text{Mayo} \langle \text{🥛} \rangle \rangle$

Options avoid these drawbacks

What is the value of incomplete languages?

 $\langle \text{Salad}(\langle \text{lettuce icon} \rangle, \circ), \text{Patty}(\langle \text{patty icon} \rangle, \text{meat icon}), \text{Sauce}(\circ, \text{mayo icon}, \text{ketchup icon}, \text{mustard icon}) \rangle$

Encoding options with choices requires context-sensitive neutral values \circ or sacrifices on sharing:

$\text{Salad}(\langle \text{burger icon} \rangle, \langle \text{lettuce icon} \rangle, \text{Patty}(\langle \text{patty icon} \rangle, \text{meat icon}), \text{Sauce}(\circ, \text{mayo icon}, \text{ketchup icon}, \text{mustard icon}) \rangle,$
 $\text{burger icon} \langle \text{lettuce icon} \rangle, \text{Patty}(\langle \text{patty icon} \rangle, \text{meat icon}), \text{Sauce}(\circ, \text{mayo icon}, \text{ketchup icon}, \text{mustard icon}) \rangle \rangle$

 $\langle \text{Salad}(\langle \text{lettuce icon} \rangle, \text{Tofu}(\langle \text{patty icon} \rangle), \text{Meat}(\langle \text{meat icon} \rangle), \text{Ketchup}(\text{ketchup icon}), \text{Mayo}(\text{mayo icon}) \rangle$

Options avoid these drawbacks but cannot express alternatives! (That is why they are incomplete.)

What is the value of incomplete languages?

 $\langle \text{Salad} \langle \text{🥬}, \circ \rangle, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$

Encoding options with choices requires context-sensitive neutral values \circ or sacrifices on sharing:

$\text{Salad} \langle \text{🍔} \langle \text{🥬}, \text{🍔} \rangle, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle,$
 $\text{🍔} \langle \text{🥬}, \text{🍔} \rangle, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Sauce} \langle \circ, \text{🥛}, \text{🍷}, \text{🥛}, \text{🍷} \rangle \rangle$

 $\langle \text{Salad} \langle \text{🥬} \rangle, \text{Tofu} \langle \text{🍔} \rangle, \text{Meat} \langle \text{🥩} \rangle, \text{Ketchup} \langle \text{🍷} \rangle, \text{Mayo} \langle \text{🥛} \rangle \rangle$

Options avoid these drawbacks but cannot express alternatives! (That is why they are incomplete.)

Conclusion: Use choices and options for completeness and usability.

 $\langle \text{Salad} \langle \text{🥬} \rangle, \text{Patty} \langle \text{🍔}, \text{🥩} \rangle, \text{Ketchup} \langle \text{🍷} \rangle, \text{Mayo} \langle \text{🥛} \rangle \rangle$

Everything is formalized in  Agda ...

... including *all* languages, definitions, theorems, and proofs,

... including tutorials, examples, and a demo,

... and where proofs come as executable compilers.

```
38 completeness-by-expressiveness :  $\forall$  {L M : VariabilityLanguage V}
```

```
39    $\rightarrow$  Complete M
```

```
40    $\rightarrow$  L  $\geq$  M
```

```
41   -----
```

```
42    $\rightarrow$  Complete L
```

```
43 completeness-by-expressiveness-enriched-in-M M M' V vs
```

```
44 ... | m , l with
```

```
45 ... | l , l'  $\rightarrow$  l = l'
```

```
46
```

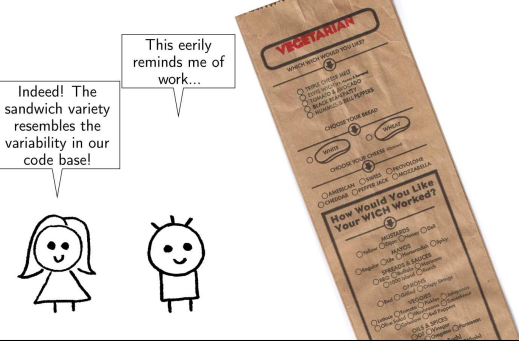
8.3k Vatrassrc/VatrassFramework/Proof/ForFree.lagda.md 32:0 14%



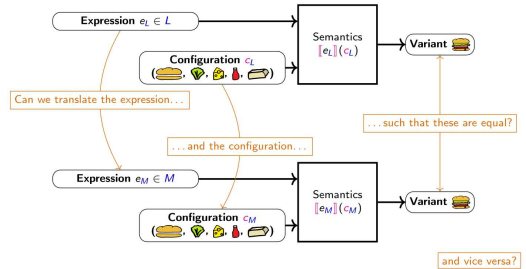
Indeed! The sandwich variety resembles the variability in our code base!

This eerily reminds me of work...

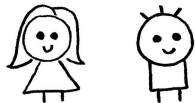




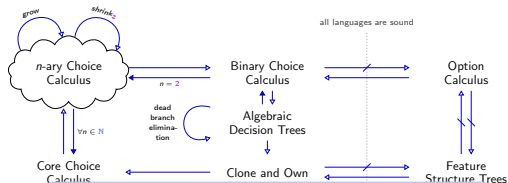
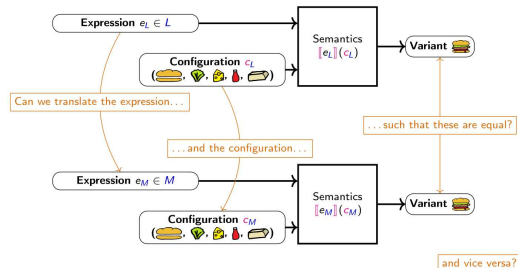
How to Compare Variability Languages L and M semantically?



This eerily
reminds me of
work...



How to Compare Variability Languages L and M semantically?

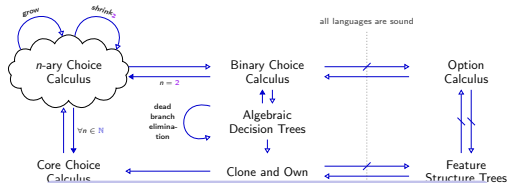
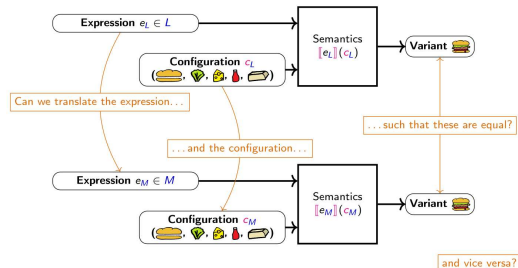


Conclusions

- Choice calculus is indeed a lambda calculus of variation as well as algebraic decision trees and clone-and-own.
- There are ≥ 3 classes of expressiveness, arising from different syntactical restrictions.



How to Compare Variability Languages L and M semantically?





Conclusions

- Choice calculus is indeed a lambda calculus of variation as well as algebraic decision trees and clone-and-own.
- There are ≥ 3 classes of expressiveness, arising from different syntactical restrictions.

Agda





 Apel, S., Kästner, C., and Lengauer, C. (2013).
Language-Independent and Automated Software Composition: The FeatureHouse Experience.
IEEE Trans. on Software Engineering (TSE), 39(1):63–79.


 Apel, S., Lengauer, C., Möller, B., and Kästner, C. (2010).
An Algebraic Foundation for Automatic Feature-Based Program Synthesis.
Science of Computer Programming (SCP), 75(11):1022–1047.

 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.


 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).


 Erwig, M. and Walkingshaw, E. (2011).
The Choice Calculus: A Representation for Software Variation.
Trans. on Software Engineering and Methodology (TOSEM), 21(1):6:1–6:27.

 Gruler, A. (2010).
A Formal Approach to Software Product Families.
PhD thesis, TU München.

 Kästner, C., Apel, S., and Kuhlemann, M. (2008).
Granularity in Software Product Lines.
In *Proc. Int'l Conf. on Software Engineering (ICSE)*, pages 311–320. ACM.

 Linsbauer, L., Lopez-Herrejon, R. E., and Egyed, A. (2017).
Variability Extraction and Modeling for Product Variants.
Software and Systems Modeling (SoSyM), 16(4):1179–1199.

 Midtgaard, J., Dimovski, A. S., Brabrand, C., and Wąsowski, A. (2015).
Systematic Derivation of Correct Variability-Aware Program Analyses.
Science of Computer Programming (SCP), 105:145–170.

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