Nullable Compositions

disjoint work with

Evan Cavallo / Carlo Angiuli
Anders Mörtberg / Andrea Vezzosi
\[ 1 + 2 = 3 \]
1 + 2 = 3 : N
floor
cubes
+
) composition
cubical TT

major difficulty: composition
for univalent universes
null compositions = no walls
Brunerie's number

a program that should output $2^*$

\[ \pi_4(S^3) \cong \mathbb{Z}/n\mathbb{Z} \]

*read Guillaume Brunerie's thesis*
(p = (i j k ((test0104 @ j) @ k) @ i))))))
i = 1)))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

2019.03.04-cubicaltt-fbdb422ada0287dbfc7b097c4a9355ed501be6e6-stack-lts9.5-brunerie2-brunerie_opt-2.output.gz
nullable compositions
nullable
not covering every corner
not “true” under double negation
not “true” under some closed substitutions
kill nullable compositions!
Plan A reduces to floor if null?
Plan A reduces to floor if null? difficult with univalence keyword: regularity
Plan B

ban nullable compositions?
Plan B

ban nullable compositions?

but universes need them in current constructions
Plan C

a different composition based on non-nullable ones

with a different set of equations to avoid regularity
Plan C

a different composition based on non-nullable ones

with a different set of equations to avoid regularity

method 1: decision tree
method 2: reflection

no general construction yet
i: I T M : A
\( i: I, j: I, k: I \vdash M: A \)
cofibrations
method 1: decision tree

\[ r_0 = r'_0, r_1 = r'_1, \ldots \]
method 1: decision tree

\[ r_0 = r', r_1 = r', \ldots \]
method 1: decision tree

\[ r_0 = r'_0, r_1 = r'_1, \ldots \]
method 1: decision tree

\[
[r_0 = r'_0, \ r_1 = r'_1, \ ...]
\]
method 1: decision tree

\[ \text{method 1: decision tree} \]

\[ \text{neo} \text{comp} \ M \quad \left[ \begin{array}{l}
\hat{r}_p = \hat{r}_p \leftrightarrow N_p, \\
\hat{r}_1 = \hat{r}_1 \leftrightarrow N_1, \ldots 
\end{array} \right] \]

See [AFH] and/or Carlo's thesis
method 1: decision tree

\[
\text{neo} \circ \text{comp } M \left[ \begin{array}{l}
\hat{r}_n = \hat{r}'_n \leftrightarrow N_n, \\
\hat{r}_1 = \hat{r}'_1 \leftrightarrow N_1, \ldots
\end{array} \right]
\]

\[
\text{comp } M \left[ \begin{array}{l}
\hat{r}_n = \hat{r}'_n \leftrightarrow \text{comp } M \left[ \begin{array}{l}
\hat{r}'_n = 0 \leftrightarrow N_n, \hat{r}'_n = 1 \leftrightarrow \text{neo} \circ \text{comp } \ldots
\end{array} \right] \\
\hat{r}_1 = 1 \leftrightarrow \text{comp } M \left[ \begin{array}{l}
\hat{r}'_1 = 1 \leftrightarrow N_n, \hat{r}'_n = 0 \leftrightarrow \text{neo} \circ \text{comp } \ldots
\end{array} \right] \\
\hat{r}_n = \hat{r}'_n \leftrightarrow N_n, \hat{r}_1 = \hat{r}'_1 \leftrightarrow N_1, \ldots
\end{array} \right]
\]
method 1: decision tree

\[
\text{neo} \text{comp } M \left[ \hat{r}_0 = \hat{r}_0' \leftrightarrow N_0, \\
\hat{r}_1 = \hat{r}_1' \leftrightarrow N_1, \ldots \right]
\]

\[
\text{comp } M \left[ \hat{r}_0 = \hat{r}_0' \leftrightarrow \text{comp } M \left[ \hat{r}_0' = 0 \leftrightarrow N_0, \hat{r}_0' = 1 \leftrightarrow \text{neo} \text{comp } \ldots \right] \\
\hat{r}_0 = 1 \leftrightarrow \text{comp } M \left[ \hat{r}_0' = 1 \leftrightarrow N_0, \hat{r}_0' = 0 \leftrightarrow \text{neo} \text{comp } \ldots \right] \\
\hat{r}_0 = \hat{r}_0' \leftrightarrow N_0, \hat{r}_1 = \hat{r}_1' \leftrightarrow N_1, \ldots \right]
\]

\[
\text{neo} \text{comp } M \left[ \right] = M
\]

See [AFH] and/or Carlo's thesis
method 1: decision tree

\[ \text{neocomp M } \left[ \begin{array}{l} r_o = r_o' \mapsto N_o, \\ r_i = r_i' \mapsto N_i, \ldots \end{array} \right] \]

limitation: the way/order to check dimension expressions needs to respect all equalities (e.g., subst.)
method 1: decision tree

variants of [AFH]-style composition

D: removal of duplicate walls
I: removal of inconsistent walls
P: permutation of walls
S: symmetry of wall constraints
\(\sigma\): symmetry for non-diagonals only
method 1: decision tree

variants of [AFH]-style composition

**DIPS**
- removal of duplicate walls
- removal of inconsistent walls
- permutation of walls
- symmetry of wall constraints
  - $\sigma$ symmetry for non-diagonals only

unsolved cases: -P+S
(no permutation, but with symmetry)
method 1: decision tree

[AFH]-style + conjunctions

$\Lambda_{\alpha} = \Lambda_{\alpha} \land \Lambda_{\beta} = \Lambda_{\beta}$
method 1: decision tree

\[ \hat{\alpha} = \alpha \wedge \hat{\gamma} = \gamma \]

trickier with +I

how about \( \hat{\gamma} = 0 \wedge \hat{\gamma} = 1 \)
**method 1: decision tree**

[AFH]-style + conjunctions

$$\land = \cdot \land \land = \cdot$$

trickier with +\| 

how about

$$\sqcap = \circ \land \sqcup = 1$$

solved case by case

[AFH], research notes, ...
method 2: reflection

[CCHM]-style composition
method 2: reflection

[CCHM]-style composition

make intervals richer so that
\[ f(r) = (r = 1) \]
is surjective
method 2: reflection

\[ \text{neo} \text{comp } M \left[ r^1 \leftrightarrow N \right] \]
method 2: reflection

\[
\text{neo}\text{comp } M \left[ \begin{array}{c} r = 1 \\ \end{array} \right] \leftrightarrow N
\]

\[
\text{comp } M \left[ \begin{array}{c} r = 1 \\ r = 0 \end{array} \right] \leftrightarrow \text{M}
\]
method 2: reflection

\text{neo}\text{comp} M \left[ \overset{r=1}{\leftrightarrow} N \right]

\text{comp} M \left[ \overset{r=1}{\leftrightarrow} N \right.
\left. \overset{r=0}{\leftrightarrow} M \right]

used in Cubical Agda
Plan C

a different composition based on non-nullable ones

with a different set of equations to avoid regularity

but, is it worth it?
none works for unknown cofibrations
none works for unknown cofibrations

we can quantify over cofibrations in cooltt

no known way to kill nullable compositions
general theory?
general theory?

build univalent Kan universes with only these cofibrations

{ \phi \mid \mathbb{Z} [\phi] } 

still very open
further reading

[Angiuli] thesis
Computational Semantics of Cartesian Cubical Type Theory

[VMA]
Cubical Agda: a dependently typed programming language with univalence and higher inductive types