absTract sOlution Analyzing Synthesis Tool (**TOAST**)

Heinz Riener
DLR, e.V.
Germany, Bremen

Rüdiger Ehlers
Uni Bremen & DFKI, GmbH
Germany, Bremen

This work was partially supported by the European Union (IMMORTAL project, grant no. 644905) and by the Institutional Strategy of the University of Bremen, funded by the German Excellence Initiative.

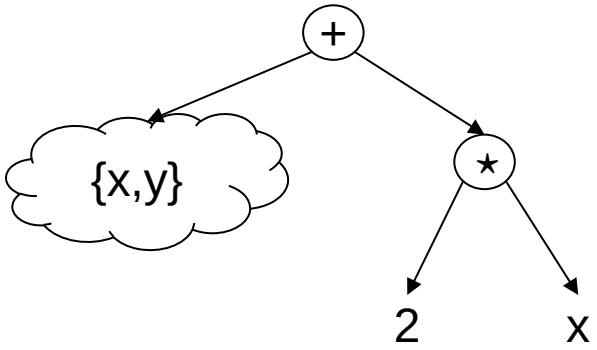
Abstract Enumeration

Grammar:
Start ::= x | y | z | (Start + Start) | (Const * Start)
Const ::= 0 | 1 | 2

Deducable concrete expressions:
x
y
z
(x + x)
(x + y)
…
(0*x)
(1*x)
…
(2*(1*x+(y+z)))
…

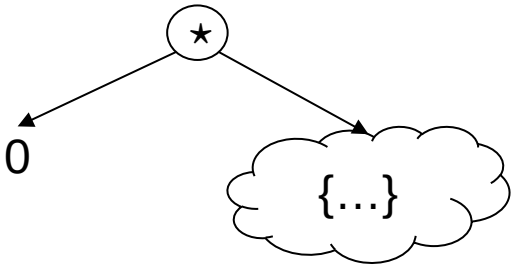
TOAST uses **abstract expressions:**

(Start{x,y} + 2*y)

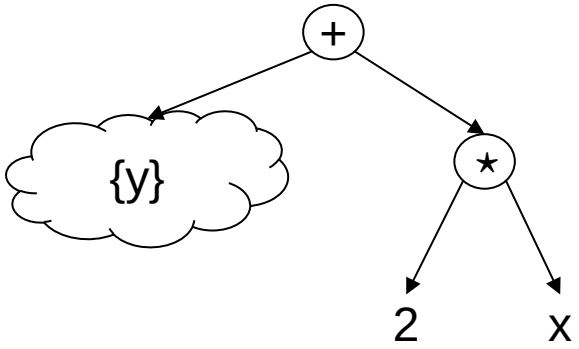


Abstract Enumeration

- Nonterminal are parametrized with support sets
- Use sanity checks to prune the search space



Discard solution if the specification requires a non-zero value for some input



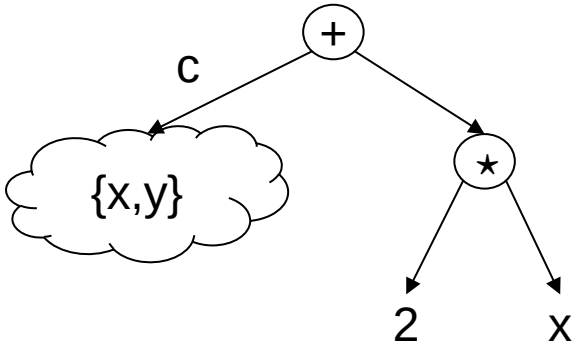
Discard solution if two inputs exists that differ only in y and must lead to the same outputs

⇨ y is not part of the solution

Functional Consistency

- SMT query to rule out abstract expression $f^{\star}$:

$$\exists i, i' \forall c, c', o, o' : \neg(Corr(i, o) \wedge \mathrm{Corr}(i', o') \wedge f^{\star}(i, c) = o \wedge f^{\star}(i', c') = o' \wedge Consistency(i, i', c, c'))$$



Abstract expression $f^{\star}(x, y) = (Start\{x, y\} + 2 \star x)$:

$$\exists x, y, x', y' \forall c, c', o, o' :$$
$$\neg(Corr(x, y, o) \wedge Corr(x', y', o') \wedge (c + (2 \star x)) = o \wedge (c' + (2 \star x')) = o' \wedge (x = x' \wedge y = y' \Rightarrow c = c'))$$

Summary

abs**T**ract s**O**lution **A**nalyzing **S**ynthesis **T**ool (**TOAST**):

- Prunes the search space by analyzing **abstract expressions**
- Several tweaks on the grammar level

- SMT queries with quantifier alternation are expensive
- Prototype implemented in Python