By Hasan Kocakaya(4914775)
Contact: h.kocakaya@student.tudelft.nl
Supervisor: Bohdan Liesnikov
Professor: Jesper Cockx
Date: 28-06-2023

Literature survey on implementation techniques for type systems: Exploring name binding techniques

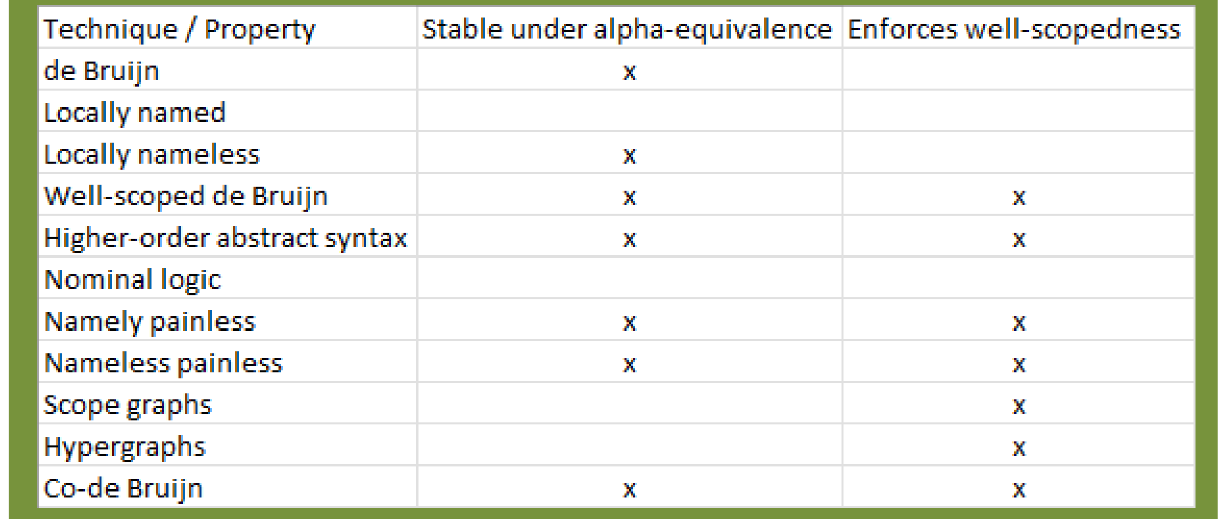
TU Delft

1. Background

- Name binding - crucial feature of type systems and programming languages.
- "Binds" a name to an entity of a program e.g variables/functions
- There are different ways to implement this
- There is no clear consensus on which technique is better/should be used.

2. Research Question

- What are the different techniques that are proposed in literature?
- What are the advantages/disadvantages of said techniques
- Is there a technique that can be identified as a one-fits-all solution?

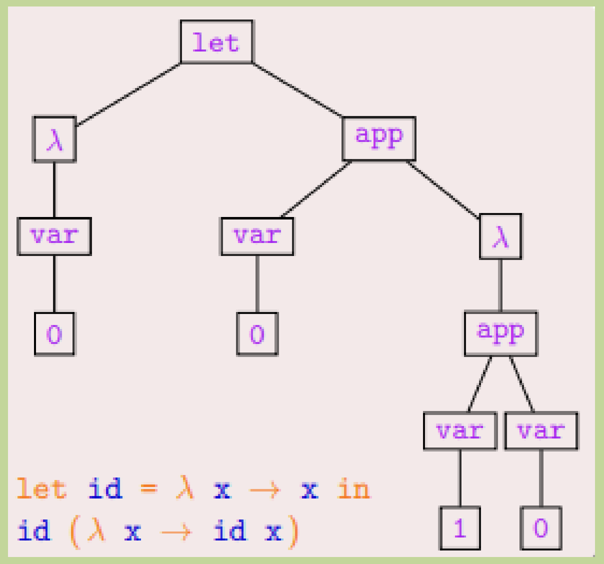


Technique / Property	Stable under alpha-equivalence	Enforces well-scopedness
de Bruijn	x	
Locally named		
Locally nameless	x	
Well-scoped de Bruijn	x	x
Higher-order abstract syntax	x	x
Nominal logic		
Namely painless	x	x
Nameless painless	x	x
Scope graphs		x
Hypergraphs		x
Co-de Bruijn	x	x

Figure 1: Overview of name-binding techniques

3. Implementation techniques

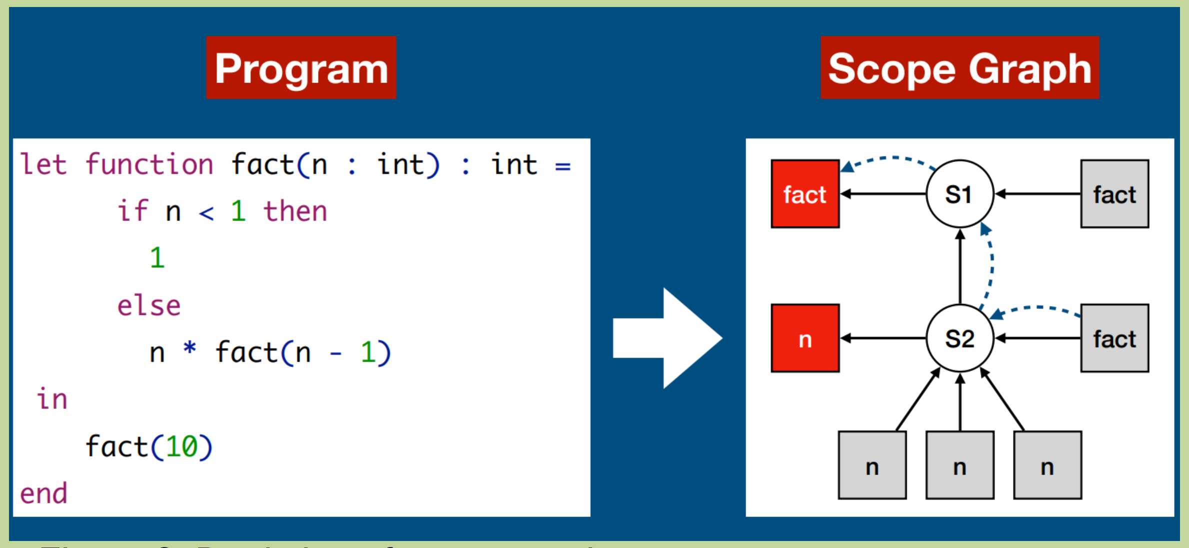
- Two examples of name-binding implementation techniques are:
- Using de Bruijn indices is the most well-known method to implement nameless name-binding.
- Variable are represented by their distance to the binding node:



```
let id = λ x → x in
id (λ x → id x)
```

Figure 2: Graphical depiction of de Bruijn indices
Safe programming with names and binders. (n.d.). [Slide show]. nicolaspouillard. https://nicolaspouillard.fr/talks/namely-painless-marburg-university.pdf

- Scope graphs represent name-binding by encoding it as a graph.
- Example program and its scope graph:



```
let function fact(n : int) : int =
      if n < 1 then
        1
      else
        n * fact(n - 1)
 in
    fact(10)
end
```

Figure 3: Depiction of scope graphs
Scope graphs: a fresh look at name-binding in programming languages. (n.d.). [Slide show]. eelcovisser. https://eelcovisser.org/talks/2017/2017-06-curryon/scope-graphs-curryon-2017-06-20.pdf

4. Comparison

- The comparison dimensions are:

1. Invariance under alpha-equivalence
2. Enforces well-scopedness
3. Ease of implementation

- Comparison dimensions chosen based on frequently occurring properties in papers
- Alpha -equivalence: two expressions are treated the exact same if the only difference is their chosen variable names.
- Well-scopedness: names can only be used when they are in scope.
- Ease of implementation: What is the complexity of the technique? Is it easy/convenient to work with?

5. Conclusion and Future Work

- No one technique is a one-fits-all solution
- Each technique has advantages and disadvantages
- Future research: Implement various name-binding techniques in the same language, for a more direct comparison.