Interleaved scope of games and automata

Murdoch J. Gabbay
Thanks to the Highlights organisers.

14 September 2017

Two kinds of scope

Two kinds of scope:

1. **Nested.** Usually this:

$$\int f(a)da, \quad \forall a.\forall b.a{=}b, \quad \text{and} \quad \lambda a.\lambda b.ab.$$

2. **Interleaved.** Usually this:

$$\texttt{alloc}\, a\ ;\ \texttt{alloc}\, b\ ;\ \texttt{do\_X}\ ;\ \texttt{dealloc}\, a\ ;\ \texttt{dealloc}\, b.$$

Above, `alloc`/`dealloc` could be (de)allocating many things: memory, a file handle, etc.

(Above, `do_X` could variously be called a variable, meta-variable, unknown, or hole. I'll call it an unknown.)

We need a more compact notation

$$\texttt{alloc}\, a\ ;\ \texttt{alloc}\, b\ ;\ \texttt{do\_X}\ ;\ \texttt{dealloc}\, a\ ;\ \texttt{dealloc}\, b$$

becomes

$$\langle a \langle b\ X\ b \rangle\, a \rangle .$$

So:

- `alloc` becomes $\langle$
- `dealloc` becomes $\rangle$
- `do_X` becomes X.
- Syntax is naturally a sequence (not a tree).
- Syntax may be composed, as in $\langle a \langle b \circ b \rangle\, a \rangle$ equals $\langle a \langle b\, b \rangle\, a \rangle$.

Nested scope is a special case of interleaved scope

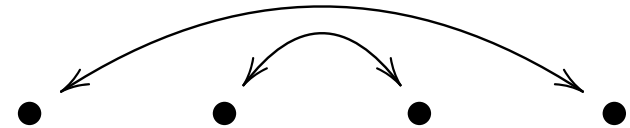
$$\lambda a.\lambda b.ab$$

becomes

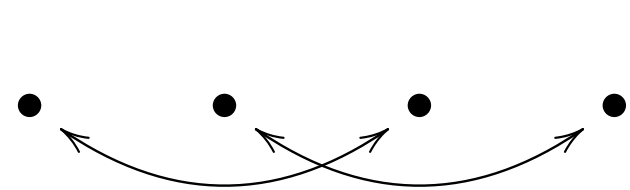
$$\lambda \langle a \lambda \langle b\, a\, b\, b \rangle\, a \rangle .$$

Two pictures: nested contrasted with interleaved

Nested (like λ) $\langle a\,\langle b\,b\rangle\,a\rangle\,=$



Interleaved (like `alloc`) $\langle a\,\langle b\,a\rangle\,b\rangle\,=$

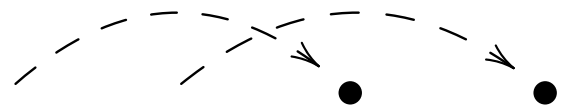


Key difference: In interleaved scope, arrows can cross, as above.

Key difference: In interleaved scope, arrows can 'dangle', as below:

$\langle a\,\langle b\,=$

$a\rangle\,b\rangle\,=$

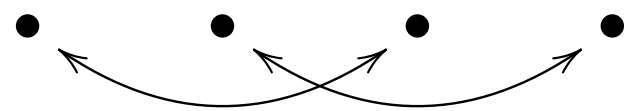


So there is a notion of time. Bindings may dangle into past and future.

Nested scope as a special case of interleaved scope

Nested (like λ) $\langle a \langle b\, b \rangle\, a \rangle =$

Interleaved (like `alloc`) $\langle a \langle b\, a \rangle\, b \rangle =$



Perhaps nested scope is to interleaved scope as groups are semigroups: important special case of more general structure.

Challenges

1. Compositional semantics of interleaved scope, including crossing and dangling arrows.
2. Development of compositional algebraic syntaxes (logics with unknowns representing diagram-fragments, whose primitive assertion is equality).

Compositionality is hard, because scope can traverse unknowns X, and interleave, and dangle into the 'future' and the 'past'.

Example questions: what do the expressions

$$\langle a\, X \quad \text{and} \quad X\, a \rangle$$

denote?

How can we α-convert a in this syntax?

Conclusions 1

Our challenge is creating syntax and semantics to interleaved scope, α-equivalence, and unknowns –

1. without insisting the user work only with ground sequences (so we don't know what's in the holes) and
2. without insisting we have complete knowledge of 'past' and 'future' (so we have truly dangling bindings).

That's the challenge.

Remarkably, it seems possible to give a clean resolution to it and to build some decently elegant mathematics around it.

Conclusions 2

I conclude with my own view of the overall development, using language from algebra:

1. Universal algebra is a logic and denotation for equality over ordinary sets and ordinary operations between them.
2. Nominal algebra is ditto, for sets with names and possibly binding operations between them.
3. This new thing is ditto, for sets with names and interleaved binding.