Skolemization, for brevity's sake

"Skolemization" refers to a technique for eliminating existential quantifiers from expressions by naming witnesses and dealing with their existence at a separate occasion. Thus we may employ the validity of

$$(*) \qquad [\, R \equiv \langle \exists x : P.x : Q.x \rangle \,]$$

as follows in a calculation:

$$\begin{array}{ll} & R \\ \equiv & \{ \bullet\ x : P.x \ , \ \text{using } (*) \} \\ & Q.x \ , \end{array}$$

with as a deferred obligation to prove that equation $x : P.x$ has a solution.

The technique is not new at all. It was brought to my attention by Lincoln A. Wallen back in 1988, but we failed to exploit it ever since. One of these days, Rik van Geldrop launched a problem that provides a stunning example of how brevity in a calculation can be achieved by applying the Skolemization technique. We will now deal with this example, once without and once with Skolemization.

* * *

The example is drawn from "language and parsing theory". We consider a grammar in which for the non-terminals Z, X, Y the only production rule for Z is

$$Z \rightarrow X\ Y \ .$$

For a non-terminal U, the set of strings belonging to the syntactic category U is denoted $\mathcal{L}(U)$. For our production rule we assume that $\mathcal{L}(X)$ and $\mathcal{L}(Y)$, and hence $\mathcal{L}(Z)$, are non-empty. Since there is only one rule for Z, we have

$$\mathcal{L}(Z) = \mathcal{L}(X) \circ \mathcal{L}(Y) \ ,$$

which is short for

(0) $$z \in \mathcal{L}(Z) \ \equiv \ \langle \exists x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) \ : \ z = x +\!\!+ \, y \rangle \qquad (\forall z) \ .$$

The problem is to design a parser for Z. To that end we define, for strings l and r of the appropriate type, predicate "parse" by

$$\text{parse}\,(Z, l, r) \ \equiv \ \langle \exists z : z \in \mathcal{L}(Z) : \ l = z +\!\!+ \, r \rangle ,$$

and we seek to derive a "recurrence relation" expressing the parse for Z in terms of parse's for X and Y.

<u>Derivation 0</u> (without Skolemization)

$$
\begin{array}{ll}
 & parse(Z, l, r) \\
\equiv & \quad \{\text{definition of } parse\} \\
 & \langle \exists z : z \in \mathcal{L}(Z) : l = z +\!\!+ r \rangle \\
\equiv & \quad \{(0) \text{ and trading}\} \\
 & \langle \exists z :: \langle \exists x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) : z = x +\!\!+ y \rangle \\
 & \qquad \wedge \; l = z +\!\!+ r \rangle \\
\equiv & \quad \{\wedge \text{ over } \exists, \text{ unnesting}\} \\
 & \langle \exists z, x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) \\
 & \qquad : z = x +\!\!+ y \; \wedge \; l = z +\!\!+ r \rangle \\
\equiv & \quad \{\text{1-point rule on } z\} \\
 & \langle \exists x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) : l = (x +\!\!+ y) +\!\!+ r \rangle \\
\equiv & \quad \{+\!\!+ \text{ is associative}\} \\
 & \langle \exists x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) : l = x +\!\!+ (y +\!\!+ r) \rangle \\
\equiv & \quad \{\text{1-point rule, introducing} \\
 & \quad \;\text{dummy } r'\} \\
 & \langle \exists r', x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) \\
 & \qquad : l = x +\!\!+ r' \; \wedge \; r' = y +\!\!+ r \rangle \\
\equiv & \quad \{\text{nesting}, \; \wedge \text{ over } \exists, \\
 & \quad \;\text{aiming at occurrences of } parse\} \\
 & \langle \exists r' :: \; \langle \exists x : x \in \mathcal{L}(X) : l = x +\!\!+ r' \rangle \\
 & \qquad \wedge \\
 & \qquad \langle \exists y : y \in \mathcal{L}(Y) : r' = y +\!\!+ r \rangle \; \rangle \\
\equiv & \quad \{\text{definition of } parse\} \\
 & \langle \exists r' :: \; parse(X, l, r') \wedge parse(Y, r', r) \rangle
\end{array}
$$

$\boxed{/\!/}$

Summarizing we arrived at

(1) $$\begin{array}{l} parse(Z, l, r) \\ \equiv \\ \langle \exists r' :: parse(X, l, r') \wedge parse(Y, r', r) \rangle \end{array}$$.

Remarks

- Derivation 0 does not use that $\mathcal{L}(X)$ and $\mathcal{L}(Y)$ are given to be non-empty. So (1) holds for empty languages as well: parse then just equivales false.
- Not all steps in the above calculation are equally fine-grained.

☒

Derivation 1 (with Skolemization)

$$\begin{array}{ll}
 & parse(Z, l, r) \\
\equiv & \quad \{ \mathcal{L}(Z) = \mathcal{L}(X) \circ \mathcal{L}(Y) \} \\
 & parse(XY, l, r) \\
 & \quad \{ \text{definition of language concatenation,} \\
 & \qquad \bullet \ x, y : x \in \mathcal{L}(X) \wedge y \in \mathcal{L}(Y) \ \} \\
 & l = (x \mathbin{+\!\!+} y) \mathbin{+\!\!+} r \\
\equiv & \quad \{ \mathbin{+\!\!+} \text{ is associative} \} \\
 & l = x \mathbin{+\!\!+} (y \mathbin{+\!\!+} r) \\
\equiv & \quad \{ \bullet \ r' : true \} \\
 & l = x \mathbin{+\!\!+} r' \ \wedge \ r' = y \mathbin{+\!\!+} r \\
\equiv & \quad \{ \bullet \ x : x \in \mathcal{L}(X) \ , \ \bullet \ y : y \in \mathcal{L}(Y) \} \\
 & parse(X, l, r') \ \wedge \ parse(Y, r', r)
\end{array}$$

☒

Summarizing, we conclude

(2) $\quad \mathit{parse}(Z, \ell, r) \equiv \mathit{parse}(X, \ell, r') \wedge \mathit{parse}(Y, r', r)$

on the condition that the "bulleted" equations are solvable (which they are by the assumption that $\mathcal{L}(X)$ and $\mathcal{L}(Y)$ are both non-empty). With the information that r' in (2) is a Skolem-variable, (1) and (2) are just equal.

Remarks

- Unfortunately, Derivation 1 relies on the non-emptiness of both $\mathcal{L}(X)$ and $\mathcal{L}(Y)$, and therefore inflicts a case-analysis upon us (be it a walk-over in this example).
- In Derivation 1, not all steps are equally fine-grained either, but notwithstanding this fact it may be clear that by Skolemization we have achieved far greater brevity and concision, without — and this should be stressed — loss of precision.

□

W.H.J. Feijen
28 April 2003