A4 Theories of strings

1. General Ideas About Strings and String Addition 2
2. A Mostly Formal Theory of Finite Strings 3
3. A Formal Theory of Finite Strings 4
4. Base-One Arithmetic 4
5. Other Definitions 5

1. General Ideas About Strings and String Addition

One can prove things about formulas and other syntactic expressions based on the premise that all such items are strings of symbols. In order to facilitate these proofs, we examine some of the basic principles about strings and string addition.

First, in addition to the string addition (infix) operator '+', we posit an infinite sequence of one-place function signs e_1, e_2, e_3, …, which are intuitively defined as follows.

$e_1(\sigma)$ = the first component of σ;
$e_2(\sigma)$ = the second component of σ;
etc.

For example, if

σ = 'abc',

then

$e_1(\sigma)$ = 'a'
$e_2(\sigma)$ = 'b'
$e_3(\sigma)$ = 'c'
$e_4(\sigma)$ = nothing
$e_5(\sigma)$ = nothing
etc.

The fundamental principle of identity for strings is *informally* given as follows.

a1:	$\alpha = \beta$ $\leftrightarrow$ $e_1(\alpha) = e_1(\beta)$ & $e_2(\alpha) = e_2(\beta)$ & …

Making this completely formal requires induction. It also requires that we construe the component identity claims as implicitly conditional – for example, '$e_2(\alpha) = e_2(\beta)$' is short for 'if either $e_1(\alpha)$ or $e_2(\alpha)$ exists, then $e_2(\alpha) = e_2(\beta)$'.

Strings divide naturally into two categories – finite and infinite – which are informally defined as follows.

a2:	A string σ is *infinite* if and only if: $e_1(\sigma)$ exists & $e_2(\sigma)$ exists & $e_3(\sigma)$ exists & … A string σ is *finite* if and only if it is not infinite.

a3:	$\forall n\{E![e_n(\sigma)] \rightarrow \forall m\{m<n \rightarrow E![e_m(\sigma)]\}\}$

Here, 'E!' is the existence predicate – $E![\tau] =_{df} \exists\nu[\nu=\tau]$.

First, Last

$$\text{first}(\sigma) =_{df} e_1(\sigma)$$

$$\forall n\{E![e_n(\sigma)] \ \&\ \sim E![e_{n+1}(\sigma)] \ .\!\rightarrow \text{last}(\sigma) = e_n(\sigma)\}$$

Examples of Theorems

t1. 'a' is finite; 'b' is finite; etc.
t2. if α and β are finite, then $\alpha+\beta$ is finite.
t3. $\alpha+(\beta+\gamma) = (\alpha+\beta)+\gamma$.
t4. $\alpha+\beta=\beta+\alpha \rightarrow \alpha=\beta$.
t5. $\alpha+\beta=\alpha+\gamma \rightarrow \beta=\gamma$.
t6. $\text{first}(\alpha+\beta) = \text{first}(\alpha)$.
t7. $\text{last}(\alpha+\beta) = \text{last}(\beta)$.

2. A Mostly Formal Theory of Finite Strings

So far, we have discussed strings both finite and infinite. Every expression in a formal language is a finite string, so it is worthwhile to consider a theory of finite strings. First, we propose a mostly formal theory; then we propose a formal theory. The first theory is formulated in a first-order language based on the following non-logical vocabulary.

(v1)	… is an atomic string	one-place predicate	$A[\alpha]$
(v2)	… is a string	one-place predicate	$S[\beta]$
(v3)	… plus …	two-place function sign	$(\alpha+\beta)$

[Note: we regard the terms 'symbol' and 'atomic string' to be synonymous; also, in programming contexts, the term 'character' is used.]

The following are axioms for this theory.

(a1) $\forall x\{Ax \rightarrow Sx\}$
(a2) $\forall x\forall y\{Sx \ \&\ Sy \ .\!\rightarrow S[x+y]\}$
(a3) $\forall x\{Ax \rightarrow \sim\exists yz\{Sy \ \&\ Sz \ \&\ x=y+z\}\}$
(a4) $\forall xyz[x+(y+z) = (x+y)+z]$
(a5) $\forall xyz[x+y = x+z \rightarrow y=z]$
(a6) $\forall xyz[x+z = y+z \rightarrow x=y]$
(a7) $\forall xy\sim\exists z[x+y+z = x]$
(a8) $\forall y\exists x_1x_2\ldots x_m\text{: } Ax_1 \ \&\ Ax_2 \ \&\ \ldots\ Ax_m \ \&\ y = x_1+x_2+\ldots+x_m$

The reason that this theory is not fully formal is axiom (a8) which is informal. This axiom intends to convey the following principle about strings.

every string is a *finite* concatenation of atomic strings.

3. A Formal Theory of Finite Strings

In order to render the theory of (finite) strings purely formal, we use the Peano technique. Furthermore, in order to render the theory as similar as possible, we introduce the concept of null or empty string, which we denote by '$\varnothing$'. The null string is the zero element in the algebra of strings.

The formal theory in question is based on the following non-logical vocabulary.

(v0)	the null string	proper noun	$\varnothing$
(v1)	… is an atomic string	one-place predicate	$A[\alpha]$
(v2)	… is a string	one-place predicate	$S[\beta]$
(v3)	… super-plus …	two-place function sign	$(\alpha^{+}\beta)$

(s1) $\varnothing$ is a string.
(s2) if σ is a string, and α is an atomic string, then $\sigma^{+}\alpha$ is a string.
(s3) nothing else is a string.
(s4) $\sim\exists xy\{Sx \,\&\, Ay \,\&\, \varnothing{=}x^{+}y\}$
(s5) $\forall xyz\{Sx \,\&\, Sy \,\&\, Az \,.\!\rightarrow x^{+}z{=}y^{+}z \rightarrow x{=}y\}$

As usual, the strict formalization of (s3) requires induction; this is given schematically as follows.

(s3*)
$$\begin{array}{l} \forall x(Ax \rightarrow \mathbb{P}[x]) \\ \& \\ \forall x\{Sx \,\&\, \mathbb{P}[x] \,.\!\rightarrow \forall y\{Ay \rightarrow \mathbb{P}[x^{+}y]\} \\ .\!\rightarrow \\ \forall x\{Sx \rightarrow \mathbb{P}[x]\} \end{array}$$

The formal theory of finite strings is analogous to the formal theory of numbers. The difference is that, numbers have single-succession, strings have multiple-succession – a string has arbitrarily-many successors, not just one.

As in the case of arithmetic, we can define addition inductively. First, the inductive definition of numerical addition goes as follows, where the variables 'm' and 'n' range over numbers.

(1) $m+0 =_{df} m$
(2) $m+n^{+} =_{df} (m+n)^{+}$

The inductive definition of string addition is analogous. The difference is that it is defined for every form of succession [every atomic string].

(b) $\sigma_1+\alpha =_{df} \sigma_1{}^{+}\alpha$
(i) $\sigma_1+(\sigma_2{}^{+}\alpha) =_{df} (\sigma_1+\sigma_2)^{+}\alpha$

Here the σ-variables range over strings, and 'α' ranges over atomic strings. Note carefully the subtle orthographic difference between '+' and '$^{+}$'. The latter is the multiple-successor operator for strings; the former is the general string addition operator.

4. Base-One Arithmetic

Evidently, Peano Arithmetic is a special case of the theory of strings. Specifically, one obtains arithmetic by restricting the class of atomic strings to just one, let us designate this string '|'. So in addition to the null string ∅, which we identify as zero, there are all the molecular strings obtained by successive applications of the one and only successor operator, which appends '|' to the string. Let us use quotes around nothing as an alternative name of the null string [as in many programming languages]. Then we have the following examples.

''	zero
'' + '|' = '|'	one
'|' + '|' = '||'	two
'||' + '|' = '|||'	three
etc.	

What we have reproduced here is, in all probability, the original formulation of arithmetic, which may be called base-one arithmetic. To use this symbolic language is quite simple One makes a mark, or uses a digit (i.e., finger), for each application of the successor operation, thereby counting (e.g., the number of mouths one must feed!)

5. Other Definitions

Although it is not crucial to the formal account of strings, we include numbers in some of the following definitions.

(f∅)	first(∅)	=	nothing		
(f+)	$\text{first}(\sigma^{+}\alpha)$	=	$\text{first}(\sigma)$, if it exists		
		=	α, otherwise		
(l∅)	last(∅)	=	nothing		
(l+)	$\text{last}(\sigma^{+}\alpha)$	=	α		
(L∅)	len(∅)	=	0		
(L+)	$\text{len}(\sigma^{+}\alpha)$	=	$\text{len}(\sigma)+1$		
(e)	$e_m(\sigma)$	=	α	if	$\sigma = \sigma_1+\sigma_2$
				and	$\text{len}(\sigma_1) = m$
				and	$\text{last}(\sigma_1) = \alpha$