Inject Code with Type Parameters

```
function addNum<START, END, ADDNUM>(
  start: START, end: END): ADDNUM {
    var addN = start + end;
    return addN;
}
```

Logical Constraints

$$E = [(\textbf{START}\ \textit{is}\ \textbf{number})\ \textit{and}\ (\textbf{END}\ \textit{is}\ \textbf{number})]$$
$$\textit{or}$$
$$[(\textbf{START}\ \textit{is}\ \textbf{string})\ \textit{and}\ (\textbf{END}\ \textit{is}\ \textbf{string})]$$

Relaxation of Logical Constraints

Continuous Space

Table 1. Probability Matrix P
$P = \begin{bmatrix} \boldsymbol{p}_1 & \dots & \boldsymbol{p}_{\mathcal{V}} \end{bmatrix}^T$.

Variable	Type		
	l_1 : **number**	l_m : **string**	$l_{\mathcal{T}}$: **any**
x_1 : **START**	$p_{1,1}$	$p_{1,m}$	$p_{1,\mathcal{T}}$
x_n : **END**	$p_{n,1}$	$p_{n,m}$	$p_{n,\mathcal{T}}$
$x_{\mathcal{V}}$: **ADDNUM**	$p_{\mathcal{V},1}$	$p_{\mathcal{V},m}$	$p_{\mathcal{V},\mathcal{T}}$

Relaxation of Logical Constraints

$$[\![E]\!]_P = \left(p_{1,1} * p_{n,1}\right) + \left(p_{1,m} * p_{n,m}\right) - \left[\left(p_{1,1} * p_{n,1}\right) * \left(p_{1,m} * p_{n,m}\right)\right]$$

Table 2. Probability Matrix P via optimizing for $[\![E]\!]_P$.

	number	**string**	**any**
START	0.05	**0.91**	0.02
END	0.06	**0.90**	0.01
ADDNUM	0.01	0.01	**0.97**

Natural Constraints

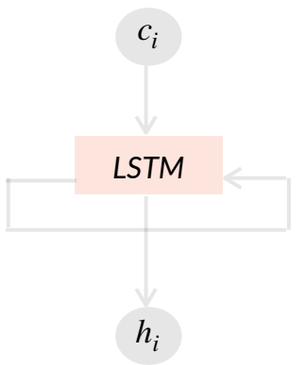

Table 3. Probability Matrix M as output of the learning model
$M = \begin{bmatrix} \boldsymbol{\mu}_1 & \dots & \boldsymbol{\mu}_{\mathcal{V}} \end{bmatrix}^T$.

	number	**string**	**any**
START	**0.39**	0.05	0.02
END	**0.36**	0.05	0.19
ADDNUM	**0.45**	0.08	0.02

Combined Constraints

$$\min_P \left(\left(\sum_v ||\boldsymbol{p}_v - \boldsymbol{\mu}_v||_2^2 \right) - \lambda [\![E]\!]_{[\boldsymbol{p}_1, \dots, \boldsymbol{p}_{\mathcal{V}}]} \right)$$

Combined Constraints

Table 4. Probability Matrix P via optimizing logical and natural constraints.

	number	**string**	**any**
START	**0.80**	0.06	0.03
END	**0.81**	0.05	0.09
ADDNUM	**0.55**	0.09	0.02

Final Typed Code

```
function addNum(start: number, end: number): number;
```