TaintStream Appendix

A DYNAMIC TRANSLATION RULES

Table 1: Examples of dynamic code translation rules. $\phi(code)$ represents applying translation rules to the code. $\langle value, tag\rangle$ is to pack a value field and a taint tag field into a structured value field (the new field name is set to the same as the value field name). V_{Col} and T_{Col} are used to get the value and tag of the column respectively. $tagMerge(tag_1, tag_2, ...)$ and $tagAgg(tag)$ are to merge multiple taint tags or aggregate a group of tags using a customized merging function. $\perp$ represents the default tag for const/insensitive values.

Index	Original	Translated
1	$\phi(source)$	$source_{tagged}$
2	$\phi(df.transformation)$	$\phi(df).\phi(transformation)$
3	$\phi(select(Col))$	$select(\phi(Col))$
4	$\phi(drop(Col))$	$drop(\phi(Col))$
5	$\phi(orderBy(Col))$	$orderBy(V_{\phi(Col)})$
6	$\phi(filter(Col))$	$filter(V_{\phi(Col)})$
7	$\phi(join(df_2, on = Col))$	$join(\phi(df_2), on = V_{\phi(Col)})$
8	$\phi(union(df_2))$	$union(\phi(df_2))$
9	$\phi(withColumn(str, col))$	$withColumn(str, \phi(col))$
10	$\phi(groupBy(Col_1)$ $.agg(F_{agg_1}(Col_2), ...))$	$groupBy(V_{\phi(Col_1)})$ $.agg(tagAgg(T_{\phi(Col_1)}), \phi(F_{agg_1}(Col_2), ...)$ $.map(x \longrightarrow (\langle x_1, x_2\rangle, x_3, x_4, ...))$
11	$\phi(map(x \longrightarrow$ $(F_i(x_{i_1}, x_{i_2}, ...),$ $F_j(x_{j_1}, x_{j_2}, ...),$ $...)))$	$map(x \longrightarrow$ $(\langle F_i(V_{x_{i_1}}, V_{x_{i_2}}, ...), tagMerge(T_{x_{i_1}}, T_{x_{i_2}}, ...)\rangle,$ $\langle F_j(V_{x_{j_1}}, V_{x_{j_2}}, ...), tagMerge(T_{x_{j_1}}, T_{x_{j_2}}, ...)\rangle,$ $...))$
12	$\phi(reduce(a, b \longrightarrow$ $(G_i(a_i, b_i), ...)))$	$reduce(a, b \longrightarrow$ $(\langle G_i(V_{a_i}, V_{b_i}), tagMerge(T_{a_i}, T_{b_i})\rangle,$ $\langle ...\rangle, ...)$
13	$\phi(column\_name))$	$column\_name$
14	$\phi(const)$	$\langle const, \perp\rangle$
15	$\phi(Func(Col_1, Col_2, ...))$	$\langle Func(V_{\phi(Col_1)}, V_{\phi(Col_2)}, ...),$ $tagMerge(T_{\phi(Col_1)}, T_{\phi(Col_2)}, ...)\rangle$
16	$\phi(F_{agg}(Col))$	$\langle F_{agg}(V_{\phi(Col)}), tagAgg(T_{\phi(Col)})\rangle$

B PROOF OF CORRECTNESS

We take the *withColumn* operation as an example to prove the correctness of our translation function ϕ, *i.e.*, the output dataframe produced by the translated *withColumn* is equivalent to the dataframe produced by the original *withColumn*, excluding the taint tags.

We first formulate some basic properties of dataframes and columns, which are regarded as the axioms in our proof.

Dataset properties. Suppose df, c, r represent a dataframe, a column, and a row respectively, we have:
(1) $df[c][r]$ represents getting the value at column c and row r.
(2) $df_1 = df_2 \Leftrightarrow \forall c, r : df_1[c][r] = df_2[c][r]$

Taint operations. Suppose $df^t = taint(df)$, we have:
(1) $df^t[c][r] = \langle df[c][r], tag\rangle$.
(2) $df^t[c][r].val = df[c][r]$.

Definition of *withColumn*. Suppose $df_2 = df_1.withColumn(str, Col)$, for $\forall$ column c and row $r \in df_2$, we have:
(1) $df_2[c][r] = df_1[c][r]$, if c's column name is not str;
(2) $df_2[c] = Col$, if c's column name is str.

We then prove the following lemma to show the correctness of the translation rules on the column, *i.e.*, *Rule* 13 $\sim$ 16.

Lemma B.1. *For $\forall df$, $df^t = taint(df)$, then for $\forall$ column c and row r, $df^t[\phi(c)][r].val = df[c][r]$.*

We prove the lemma using mathematical induction:

(1) if c is column_name, getting a column by its name will return a tainted column, so $df^t[\phi(c)][r].val = df[c][r]$.

(2) else if c is const, according to the Rule 14, $\phi(c) = \langle const, \perp\rangle$, so $df^t[\phi(c)][r].val = df[c][r]$.

(3) else if c is $Func(c_1, c_2, ...)$, we have:

$$\begin{aligned}
\phi(c) &= \phi(Func(c_1, c_2, ...)) \\
&= \langle Func(V_{\phi(c_1)}, V_{\phi(c_2)}, ...), tagMerge(tag_1, tag_2, ...)\rangle \\
&\qquad \text{.............. Rule 15} \\
&= \langle Func(V_{\phi(c_1)}, V_{\phi(c_2)}, ...), tag\rangle \text{ .... tag merge} \\
&= \langle Func(c_1, c_2, ...), tag\rangle \text{ .... mathematical induction} \\
&= \langle c, tag\rangle \text{ .... } c \text{ is } Func(c_1, c_2, ...)
\end{aligned}$$

so $df^t[\phi(c)][r].val = df^t[\langle c, tag\rangle][r].val = df[c][r]$

(4) else if c is $F_{agg}(c)$, the proof is similar to (3).

Overall, we have proved $df^t[\phi(c)][r].val = df[c][r]$.

Now we prove the correctness of the translation rule of *withColumn*. Specifically, we need to prove that the output dataframe through the translated *withColumn* operation is equivalent to the dataframe through the original *withColumn* operation, excluding the taint tags, which can be formalized as follows.

Proof. For $\forall df_1$, $df_1^t = taint(df_1)$; and for $\forall str$, Col, $df_2 = df_1.withColumn(str, Col)$. Then for $\forall c, r, \phi(df_2)[c][r].val = df_2[c][r]$.

$$\begin{aligned}
\phi(df_2) &= \phi(df_1.withColumn(str, Col)) \\
&= \phi(df_1).\phi(withColumn(str, Col)) \text{ .... Rule 2} \\
&= df_1^t.withColumn(str, \phi(Col)) \text{ ..... Rule 1, Rule 9} \\
&= df_1^t.withColumn(str, \langle Col, tag\rangle) \text{ .... Lamma B.1}
\end{aligned}$$

If c's column name is str, then
$\phi(df_2)[c][r].val = \langle Col, tag\rangle.val = Col[r] = df_2[c][r]$
Else, c's column name is not str, then
$\phi(df_2)[c][r].val = df_1^t[c][r].val = df_1[c][r] = df_2[c][r]$
Overall, we have prove $\phi(df_2)[c][r].val = df_2[c][r]$ □