Transaction Systems
Exercise Session 2

Andrey Gubichev

October 28, 2013
Formally prove that all the schedules generated by the Strict2PL scheduler are recoverable.

List the properties of the following schedules:

- \( H_1 = w_1[x]w_2[x]w_2[y]w_1[y]c_2c_1 \) (RC, ACA)
- \( H_2 = w_1[x]r_2[y]r_1[x]c_1 r_2[x]w_2[y]c_2 \) (SR, RC, ACA, ST)
- \( H_3 = w_1[x]r_2[y]r_1[x]r_2[x]c_1 w_2[y]c_2 \) (SR, RC)
- \( H_4 = w_1[x]r_2[y]r_2[x]r_1[x]c_2 w_1[y]c_1 \) (-)
Today’s plan

- Herbrand semantics
- Final state serializability
- View serializability
Read operation $r_i(x)$ reads the value written by the last $w_j(x)$ before $r_i(x)$

Write operation $w_i(x)$ writes a new value that depends on everything that the transaction $t_i$ has read so far
Herbrand semantics: formalism

- $H_s(r_i(x)) = H(w_j(x))$, $w_j(x)$ is the last write before $r_i$
- $H_s(w_i(x)) = f_{ix}(H_s(r_i(y_1)), \ldots, H_s(r_i(y_m)))$, where $r_i(y_j)$ are all read operations of $t_i$ before $w_i(x)$
- $f_{ix}$ is an uninterpreted $m$-ary function symbol
Herbrand semantics: formalism

- $H_s(r_i(x)) = H(w_j(x))$, $w_j(x)$ is the last write before $r_i$
- $H_s(w_i(x)) = f_{ix}(H_s(r_i(y_1)), \ldots, H_s(r_i(y_m)))$, where $r_i(y_j)$ are all read operations of $t_i$ before $w_i(x)$
- $f_{ix}$ is an uninterpreted $m$-ary function symbol
- every schedule starts with $t_0$: writing all the data items

\[
s = r_1(x)r_2(y)\ldots
\]
Herbrand semantics: formalism

- $H_s(r_i(x)) = H(w_j(x))$, $w_j(x)$ is the last write before $r_i$
- $H_s(w_i(x)) = f_{ix}(H_s(r_i(y_1)), \ldots, H_s(r_i(y_m)))$, where $r_i(y_j)$ are all read operations of $t_i$ before $w_i(x)$
- $f_{ix}$ is an uninterpreted $m$-ary function symbol
- every schedule starts with $t_0$: writing all the data items

$s = w_0(x)w_0(y)c_0r_1(x)r_2(y)\ldots$
Herbrand semantics: example

\[ s = w_0(x)w_0(y)c_0r_1(x)r_2(y)w_2(x)w_1(y)c_2c_1 \]

\[ H_s(w_0(x)) = f_{0x}(\) \]

\[ H_s(w_0(y)) = f_{0y}(\) \]
Herbrand semantics: example

\[ s = w_0(x)w_0(y)c_0r_1(x)r_2(y)w_2(x)w_1(y)c_2c_1 \]

- \( H_s(w_0(x)) = f_0x() \)
- \( H_s(w_0(y)) = f_0y() \)
- \( H_s(r_1(x)) = H_s(w_0(x)) = f_0x() \)
- \( H_s(r_2(y)) = H_s(w_0(y)) = f_0y() \)
Herbrand semantics: example

\[ s = w_0(x) w_0(y) c_0 r_1(x) r_2(y) w_2(x) w_1(y) c_2 c_1 \]

- \( H_s(w_0(x)) = f_{0x}() \)
- \( H_s(w_0(y)) = f_{0y}() \)
- \( H_s(r_1(x)) = H_s(w_0(x)) = f_{0x}() \)
- \( H_s(r_2(y)) = H_s(w_0(y)) = f_{0y}() \)
- \( H_s(w_2(x)) = f_{2x}(H_s(r_2(y))) = f_{2x}(f_{0y}()) \)
- \( H_s(w_1(y)) = f_{1y}(H_s(r_1(x))) = f_{1y}(f_{0x}()) \)
Herbrand semantics: schedule

\[ H[s](x) = H(w_j(x)), \] 
where \( w_j(x) \) is the last operation writing \( x \), for each \( x \)
Example

\[ s = w_0(x)w_0(y)c_0r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2w_1(y)c_1 \]
Final State Serializability

Two schedules $s$ and $s'$ are final state equivalent, iff

$op(s) = op(s')$ and $H[s] = H[s']$
Example

\[ s = w_0(x)w_0(y)c_0r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2w_1(y)c_1 \]

equivalent to

\[ t_2t_1 \]
One more extension to the schedule

\[ s = w_0(x)w_0(y)c_0 r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2 w_1(y)c_1 r_{\infty}(x)r_{\infty}(y)c_{\infty} \]
The reads-from relation of \( s \):

\[
\text{RF}(s) = \{(t_i, x, t_j) \mid r_j(x) \text{ reads } x \text{ from } w_i(x)\}
\]
Alive steps

- Step $p$ is directly useful for step $q$ ($p \rightarrow q$), if
  - $q$ reads from $p$
  - or $p$ is read and $q$ is subsequent write in the same transaction
- Transitive closure of $\rightarrow$: $\rightarrow^*$
- A step $p$ is alive if it is useful for some step from $t_\infty$
Example

\[ s = r_1(x)r_2(y)w_1(y)w_2(y)c_1 c_2 r_\infty(x)r_\infty(y)c_\infty \]
Example

\[ s = r_1(x) r_2(y) w_1(y) w_2(y) c_1 c_2 r_\infty(x) r_\infty(y) c_\infty \]

- \( w_2(y) \) is alive
- \( r_2(y) \) is alive
- \( r_1(x) \) is not alive
Live Reads-from

The *live reads-from* relation of $s$:

$$LRF(s) = \{(t_i, x, t_j) \mid \text{an alive } r_j(x) \text{ reads } x \text{ from } w_i(x)\}$$
Example

\[ s = w_0(x)w_0(y)c_0r_1(x)r_2(y)w_1(y)w_2(y)c_1c_2r_\infty(x)r_\infty(y)c_\infty \]
Example

\[ s = w_0(x)w_0(y)c_0r_1(x)r_2(y)w_1(y)w_2(y)c_1c_2r_\infty(x)r_\infty(y)c_\infty \]

- \( RF(s) = \{(t_0, x, t_1), (t_0, y, t_2), (t_0, x, t_\infty), (t_2, y, t_\infty)\} \)
- \( LRF(s) = \{(t_0, y, t_2), (t_0, x, t_\infty), (t_2, y, t_\infty)\} \)
Two schedules are FSR equivalent iff

- $op(s) = op(s')$
- $LRF(s) = LRF(s')$
View serialization

Two schedules are view equivalent, iff

\[ \text{op}(s) = \text{op}(s') \]
\[ H[s] = H[s'] \]
\[ H_s(p) = H_{s'}(p) \text{ for every read or write step } p \]
VSR criteria

Two schedules are *view equivalent*, iff

- \( \text{op}(s) = \text{op}(s') \)
- \( \text{RF}(s) = \text{RF}(s') \)
Example

\[ s = w_0(x)w_0(y)c_0r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2w_1(y)c_1 \]
$$s = w_0(x)w_0(y)c_0 r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2 w_1(y)c_1$$

- $RF(s) = 
  \{(t_0, x, t_1), (t_1, x, t_2), (t_0, y, t_2), (t_1, x, t_\infty), (t_1, y, t_\infty)\}$

- $LRF(s) = \{(t_0, x, t_1), (t_1, x, t_\infty), (t_1, y, t_\infty)\}$
Example

\[ s = w_0(x)w_0(y)c_0r_1(x)w_1(x)r_2(x)r_2(y)w_2(y)c_2w_1(y)c_1 \]

- \( RF(s) = \{(t_0, x, t_1), (t_1, x, t_2), (t_0, y, t_2), (t_1, x, t_\infty), (t_1, y, t_\infty)\} \)
- \( LRF(s) = \{(t_0, x, t_1), (t_1, x, t_\infty), (t_1, y, t_\infty)\} \)
- not in VSR (but in FSR)
Given the schedule

\[ s = r_1(x)r_3(x)w_3(y)w_2(x)c_3r_4(y)w_4(x)c_2r_5(x)c_4w_5(z)w_1(z)c_1c_5 \]

For some transactions, the function computed by the write step is the identity function in one of its arguments (copier):

\[ f_{ix}(v_1, \ldots, v_m) = v_j \text{ for some } j \]

Compute the Herbrand semantics of \( s \) given that \( t_3 \) and \( t_4 \) are copiers.
Given the schedule

\[ s = r_1(x)r_3(x)w_3(y)w_2(x)r_4(y)c_2w_4(x)c_4r_5(x)c_3w_5(z)c_5w_1(z)c_1 \]

Does it belong to FSR? VSR? CSR?
Info

- Exercises due: 9 AM, November 4, 2013
- Submit to andrey.gubichev@in.tum.de