Today’s Plan

- ERRATA: Queue commutativity table (slide 6.33) lower left: ”impossible” (Thanks Ahmet)
- Last week’s homework
- Escrow locking
- Transaction Chopping
- Homework
Escrow Locking

- Credits: Dr. Andrey Gubichev, 2013
Escrow locking

- For counter objects: there are bounds $low$, $high$ and current possible value range $sup$, $inf$

- incr ($x$, $D$):
  
  if $x.sup + D \leq x.high$ then
  
  $x.sup := x.sup + D$; return ok

  else if $x.inf + D > x.high$ then
  
  return no

  else wait fi fi;

- decr ($x$, $D$):
  
  if $x.low \leq x.inf - D$ then
  
  $x.inf := x.inf - D$; return ok

  else if $x.low > x.sup - D$ then
  
  return no

  else wait fi fi;
Escrow locking

- When committing: adjust $inf$, $sup$
- When aborting: roll back $inf$, $sup$
- Commit of transaction $t$:
  for each operation $\text{incr}(x, D)$ executed by $t$ do
  $x.\inf := x.\inf + D$
  od;
  for each operation $\text{decr}(x, D)$ executed by $t$ do
  $x.\sup := x.\sup - D$
  od;
- Abort of transaction $t$:
  for each operation $\text{incr}(x, D)$ executed by $t$ do
  $x.\sup := x.\sup - D$
  od;
  for each operation $\text{decr}(x, D)$ executed by $t$ do
  $x.\inf := x.\inf + D$
Escrow Locking

- Credits: Dr. Andrey Gubichev, 2013
Escrow Locking

Consider two counting objects $x$ and $y$, with initial values $x = 100$ and $y = 50$. Both counters have zero as lower bound and no upper bound. Apply the escrow locking method to the following schedule of three transactions, one of which aborts:

$decr_1(x, 60)\hspace{1em} incr_2(x, 20)\hspace{1em} incr_1(x, 10)\hspace{1em} decr_3(x, 50)$

$decr_2(y, 60)\hspace{1em} incr_2(x, 20)\hspace{1em} a_2\hspace{1em} decr_1(y, 10)\hspace{1em} c_1\hspace{1em} c_3$
Transaction Chopping

- Credits: Dr. Andrey Gubichev, 2013
Transaction chopping

- Split transactions into sub-transactions that run (potentially) concurrently
- Chopping is correct if every execution of the transaction pieces is conflict equivalent to some serial history of original transactions
- Rule 1: precedence from the original schedule is kept
- Rule 2: each piece is executed according to some CC protocol
Consider the following transactions:

- \( t_1 = r_1(x) \ w_1(x) \ r_1(y) \ w_1(y) \)
- \( t_2 = r_2(x) \)
- \( t_3 = r_3(y) \ w_3(y) \)

Try to decompose \( t_1 \) into three pieces such that the result is a correct chopping.
Transaction Chopping

► Credits: Dr. Andrey Gubichev, 2013
Next Week’s Lecture

- Exam relevance of chapters/algorithms/...
- Answers to your questions
  (send e-mails!)
- Practicing old exam questions
  (bring your notes and computers for help (not allowed in the exam!))
Homework

- Last homework!
- Already uploaded to our website.