CS 337 **Solution to Quiz 2** **11/3/04**

1. (Rabin-Karp algorithm for string search)

   (a) q is not required to be prime. Given that d and q are powers of 2: in computing $((u - a \times f) \times d + b) \bmod q$, multiplication by d is a left shift and mod q can be computed by a right shift.

   (b) After choosing q (and d), make one pass over the text to compute $val(p)$ for all substrings p of length 20, and enter them in a hash table along with a pointer to p. To match a pattern, let r be its prefix of length 20. Compute $val(r)$ and look it up in the hash table. For all occurrences of this value, match the pattern starting at the beginning of the corresponding substring in the text.

2. (KMP algorithm for string search)

   (a) Recall that $u \preceq v \;\equiv\; u \sqsubseteq v \wedge u \;\leq\; v$.

   - $u \preceq u$:

$$
\begin{array}{ll}
 & u \preceq u \\
= & \{\text{definition}\} \\
 & u \sqsubseteq u \wedge u \;\leq\; u \\
= & \{\sqsubseteq \text{ and } \leq \text{ are partial orders}\} \\
 & \mathit{true} \wedge \mathit{true} \\
= & \{\text{simplify}\} \\
 & \mathit{true}
\end{array}
$$

   - $u \preceq v \;\wedge\; v \preceq u \;\Rightarrow\; u = v$:

$$
\begin{array}{ll}
 & u \preceq v \;\wedge\; v \preceq u \\
\Rightarrow & \{\text{definition}\} \\
 & u \sqsubseteq v \;\wedge\; v \sqsubseteq u \\
\Rightarrow & \{\sqsubseteq \text{ is a partial order}\} \\
 & u = v
\end{array}
$$

   - $u \preceq v \;\wedge\; v \preceq w \;\Rightarrow\; u \preceq w$:

$$
\begin{array}{ll}
 & u \preceq v \;\wedge\; v \preceq w \\
\Rightarrow & \{\text{definition}\} \\
 & u \sqsubseteq v \wedge u \;\leq\; v \wedge v \sqsubseteq w \wedge v \;\leq\; w \\
\Rightarrow & \{\text{Rearranging terms}\} \\
 & u \sqsubseteq v \wedge v \sqsubseteq w \wedge u \;\leq\; v \wedge v \;\leq\; w \\
\Rightarrow & \{\sqsubseteq \text{ and } \leq \text{ are partial orders}\} \\
 & u \sqsubseteq w \wedge u \;\leq\; w \\
\Rightarrow & \{\text{definition}\} \\
 & u \preceq w
\end{array}
$$

   (b) The only u that satisfies $c(v) \preceq u \prec v$ is $c(v)$. For $v =$ "ababab", $c(v)$ is "abab".

   (c) It is possible that *both* $p[\overline{u}] = p[\overline{v}]$ and $u = \epsilon$ hold on termination of the loop. Consider $p =$ "aa". Let $v =$ "a"; then $\overline{v} = 1$, $u = c(v) = \epsilon$, $\overline{u} = 0$. We have $p[0] = p[1] \wedge u = \epsilon$.

	Title	Actor
R	Men in Black	Will Smith
S	Men in Black	Tommy Lee Jones

Table 1: Two relations

3. (Relational Algebra)

   (a) The names of theatres which are showing G-rated movies in which Audrey Hepburn is acting, is given by

   $$\pi_{Theatre}(\sigma_{p \wedge q}\langle R \bowtie S\rangle)$$

   where

   p be $Actor =$ Audrey Hepburn
   q be $Rating =$ G

   (b) Consider relations R and S in table 1. Each relation has two attributes, and just one tuple.

   Take attribute a to be Title. Now, $R \cap S$ is empty, so $\pi_a(R \cap S)$ is also empty. However, $\pi_a(R)$ and $\pi_a(S)$ both have a single row (and single column) with the entry "Men in Black". So, $\pi_a(R) \cap \pi_a(S)$ has one row.

4. (Powerlist; 15 points)

   (a) $swap\langle x\rangle = \langle x\rangle$
   $swap(p \bowtie q) = q \bowtie p$

   (b) Proof is by induction. For singleton list $\langle x\rangle$, both sides are $\langle x\rangle$. Consider a list of the form $(p \bowtie q)$.

$$
\begin{array}{cl}
 & rev(rr(p \bowtie q)) \\
= & \{\text{definition of } rr\} \\
 & rev(q \bowtie rr(p)) \\
= & \{\text{definition of } rev\} \\
 & rev(rr(p)) \bowtie rev(q) \\
= & \{\text{induction: } rev(rr(p)) = lr(rev(p))\} \\
 & lr(rev(p)) \bowtie rev(q) \\
= & \{\text{definition of } lr\} \\
 & lr(rev(q) \bowtie rev(p)) \\
= & \{\text{definition of } rev\} \\
 & lr(rev(p \bowtie q))
\end{array}
$$