Given: $M = (Q, Q_0, \delta)$ is a state machine.
Prove: P is an invariant of M.
1. $\forall q \in Q_0 (P(q))$
2. $\forall (q, q') \in \delta (P(q) \Rightarrow P(q'))$.
3. QED Invariant theorem

Given: $M = (Q, Q_0, \delta)$ is a state machine.
Given: $\forall q \in Q_0(P(q))$
Given: $\forall (q, q') \in \delta(P(q) \Rightarrow P(q'))$.
Prove: P is an invariant of M, that is, $\forall q \in Q(q$ reachable $\Rightarrow P(q)$
1. $\forall n \geq 0(P'(n))$, where $P'(n) \equiv$ "$\forall q$, a final state of an n-step exe
    1. (Base) $P'(0)$ By the assumption about start
    2. (Inductive step) $\forall n \geq 0(P'(n) \Rightarrow P'(n+1))$
        1. Fix $n \geq 0$
        2. Assume $P'(n)$
        3. $P'(n+1)$???
        4. QED
    3. QED Induction
2. QED Definition of "reachable"

3. $P'(n+1)$, that is, P is true for all final states of $n+1$-step exe

   1. Fix an $n+1$-step execution $q_0, q_1, \ldots, q_n, q_{n+1}$.
   2. q_n is the final state of an n-step execution.
   3. $P(q_n)$ Inductive hypothesis (1.2.2)
   4. (q_n, q_{n+1}) is a step of M.
   5. $P(q_{n+1})$ By the assumption about steps preser
   6. QED UG

Prove: P is an invariant of the Die Hard state machine.

1. $\forall q \in Q_0(P(q))$ — P true for start state $(0,0)$.
2. $\forall (q,q') \in \delta(P(q) \Rightarrow P(q'))$.
    1. Fix $(q,q') \in \delta$
    2. Assume $P(q)$.
    3. $P(q')$
        1. If (q,q') fills the little jar, then $P(q')$.
        2. If (q,q') fills the big jar, then $P(q')$.
        3. If (q,q') empties the little jar, then $P(q')$.
        4. If (q,q') empties the big jar, then $P(q')$.
        5. If (q,q') pours from little to big, then $P(q')$.
        6. If (q,q') pours from big to little, then $P(q')$.
        7. QED Cases
    4. QED — Implication, UG
3. QED — Invariant theorem

Prove: P is an invariant of the strong caching state machine.

1. $\forall q \in Q_0(P(q))$ — P when both caches are empty.
2. $\forall (q, q') \in \delta(P(q) \Rightarrow P(q'))$.
    1. Fix $(q, q') \in \delta$
    2. Assume $P(q)$.
    3. $P(q')$
        1. If (q, q') puts a new value in memory, then $P(q')$.
           Makes the caches null for the a
        2. If (q, q') updates a cache from memory, then $P(q')$.
           Makes cache = memory, at the
        3. If (q, q') drops a value from a cache, then $P(q')$.
           Makes cache null.
        4. QED — Cases
    4. QED
3. QED — Invariant theorem

Prove: P is an invariant of the matching state machine.

1. $\forall q \in Q_0(P(q))$ — Vacuously true—everyone pursuing
2. $\forall (q, q') \in \delta(P(q) \Rightarrow P(q'))$.
   1. Fix $(q, q') \in \delta$
   2. Assume $P(q)$.
   3. $P(q')$ — ???
   4. QED
3. QED — Invariant theorem

3. $P(q')$, that is, $\forall b, g((b$ has crossed off g in $q')$
$\Rightarrow (g$ has a pursuer she prefers over b in

   1. Fix Bob, Gail.
   2. Assume Bob has crossed off Gail, in q'.
   3. Gail has a pursuer she prefers over Bob, in q'.
      1. If in (q, q'), Gail rejects Bob, then Gail has a pursuer she over Bob, in q'. Definition of the ste
      2. If in (q, q'), another girl rejects Bob, then Gail has a purs prefers over Bob, in q'. ???
      3. If in (q, q'), Gail rejects another boy, then Gail has a purs prefers over Bob, in q'. ???
      4. If in (q, q'), another girl rejects another boy, then Gail ha prefers over Bob, in q'. Doesn't affect the tr
      5. QED Cases
   4. QED Implication, UG

2. If another girl rejects Bob, then Gail has a pursuer she prefers o
   1. Assume another girl rejects Bob in (q, q').
   2. Bob has crossed off Gail in q. 2.3.2, doesn't cross off Ga
   3. Gail has a pursuer she prefers over Bob, in q.
      By assumption $P(q)$ (2.2)
   4. Gail has a pursuer she prefers over Bob, in q'.
      Gail's pursuers don't chang
   5. QED Implication

3. If Gail rejects another boy, then Gail has a pursuer she prefers
   1. Assume Gail rejects another boy in (q, q').
   2. Bob has crossed off Gail in q. 2.3.2, doesn't move in this
   3. Gail has a pursuer she prefers over Bob, in q.
      $P(q)$ (2.2).
   4. Gail has a pursuer she prefers over Bob, in q'.
      Gail doesn't send away top
   5. QED Implication

Given: $M = (Q, Q_0, \delta)$ is a state machine.
Given: e is an execution of M, $k \in N$.
Prove: e has at most k steps.
1. $f : Q \to N$ is a termination function for M.
2. $f(q) \leq k$, where $q \in Q_0$ is the first state of e.
3. QED Time bound theorem

Given: $M = (Q, Q_0, \delta)$ is a state machine.
Prove: Every execution of M has at most k steps.
1. $f : Q \to N$ is a termination function for M.
2. $\forall q \in Q_0 (f(q) \leq k)$
3. QED Time bound theorem

Given: $M = (Q, Q_0, \delta)$ is a state machine.
Prove: Every execution of M is finite.
1. $f : Q \to N$ is a termination function for M.
2. QED Termination theorem

Given: M is the matching state machine.
Given: $f(q)$ is defined as $\Sigma_b(n - \textit{next-number}(b))$.
Prove: Every execution of M has at most $n(n-1)$ steps.

1. f is a termination function for M.
   1. f is defined on all reachable states. Invariant 1
   2. $\forall q, q'$, reachable states with $(q, q') \in \delta(f(q') < f(q))$
      Definition of step.
2. $\forall q \in Q_0(f(q) \leq n(n-1))$ It's equal: all boys start
3. QED Time bound theorem