DISCTRU Machine Project

Due on **April 2, 2017**

You are to implement a computer program following the specifications of a system given below.

- Applicable Sets

  $\mathbf{P} : \{1,2,3\} \times \{1,2,3\}$
  $\mathbf{B} : \{\mathsf{true}, \mathsf{false}\}$
  $\mathbf{F} : \{1,3,5,7,9\}$
  $\mathbf{S} : \{0,2,4,6,8\}$
  $\mathbf{N}$: set of natural numbers

  $\mathbf{J} : \{(1,1),\ (1,2),\ (1,3)\}$
  $\mathbf{K} : \{(3,1),\ (3,2),\ (3,3)\}$
  $\mathbf{L} : \{(1,1),\ (2,2),\ (3,3)\}$
  $\mathbf{M} : \{(3,1),\ (2,2),\ (1,3)\}$

- System Variables

  $tposn, free \subseteq \mathbf{P}$
  $okay, uno, dos, tres, quad \subseteq \mathbf{N}$

  $turn, over, good, nice \in \mathbf{B}$

- System Facts

  $free = \mathbf{P} - (tposn)$

  $okay = \mathbf{N} - (uno \cup dos \cup tres \cup quad)$

- System Initialization

  $over = \mathsf{false}$
  $turn = \mathsf{true}$
  $good = \mathsf{false}$
  $nice = \mathsf{false}$

  $tposn = \varnothing$
  $uno = \varnothing$
  $dos = \varnothing$
  $tres = \varnothing$
  $quad = \varnothing$

- System States and Behavior

  $\texttt{Count(}moves \subseteq \mathbf{N}\texttt{)} = \sum_{v \in moves} v$

  $\texttt{Check(}moves \subseteq \mathbf{N}\texttt{)} = |moves| = 3 \wedge \texttt{Count}(moves) = 15$

  `NextPlayerMove(`$posn \in \mathbf{P}, move \in \mathbf{N}$`):`

  $(move \in okay \wedge (turn \wedge move \in \mathbf{F} \vee \neg turn \wedge move \in \mathbf{S})) \rightarrow good = \mathsf{true}$
  $(good \wedge posn \in free \wedge posn \in \mathbf{J}) \rightarrow uno = uno \cup \{move\}$
  $(good \wedge posn \in free \wedge posn \in \mathbf{K}) \rightarrow dos = dos \cup \{move\}$
  $(good \wedge posn \in free \wedge posn \in \mathbf{L}) \rightarrow tres = tres \cup \{move\}$
  $(good \wedge posn \in free \wedge posn \in \mathbf{M}) \rightarrow quad = quad \cup \{move\}$
  $(good \wedge posn \in free \wedge (\texttt{Check}(uno) \vee \texttt{Check}(dos) \vee \texttt{Check}(tres) \vee \texttt{Check}(quad))) \rightarrow (over = \mathsf{true} \wedge nice = \mathsf{true})$
  $(\neg over \wedge good \wedge posn \in free) \rightarrow (turn = \neg turn \wedge tposn = tposn \cup \{posn\})$
  $(\neg over \wedge |free| = 0) \rightarrow over = \mathsf{true}$
  $good \rightarrow good = \mathsf{false}$

  `EndGame(over):`

  $result \in \{$ first, second, draw $\}$
  $(over \wedge nice \wedge turn) \rightarrow result =$ first
  $(over \wedge nice \wedge \neg turn) \rightarrow result =$ second
  $(over \wedge \neg nice) \rightarrow result =$ draw