Event Analytics and Verification: PAT Approach

Jin-Song Dong (first met Ed at Marktoberdorf 1994)
National University of Singapore (NUS)
PAT team: two former PhD students: prof. J. Sun (SUTD), prof. Y. Liu (NTU) and
20+ PhD/Postdocs
PAT System - based on Hoare’s \textit{event} based formalism CSP
(CAV’09’12’13’14, FM’11’12’14, TOSEM’13’14, TSE’13’14, FMSD’14)

Support event based formalisms:
CSP\# with shared variables, Timed-CSP, Probabilistic-CSP …
Model checking as planning/scheduling [FMSD’14]

```c
//@@Sliding Game@@
//The following models the sliding game with the extra 'costs' complexity

var board[9]:{0..8} = [3,5,6, 0,1,2 :index
                      0,2,7, 3 4 5 :index
                      8,4,1]; // 6,7,8 :index
hvar empty:{0..8} = 3; //empty position is a secondary variable, no need to put it in the state space

var c = 0; // cost utility, e.g. costs 1 for left and right move, 2 for up, 0 for down

Game() = Left() [] Right() [] Up() [] Down();

Left() = [empty!=2 && empty!=5 && empty!=8] left
          {board[empty]=board[empty+1]; board[empty+1]=0; empty=empty+1; c++} -> Game();

Right() = [empty!=0 && empty!=3 && empty!=6] right
           {board[empty]=board[empty-1]; board[empty-1]=0; empty=empty-1; c++} -> Game();

Up() = [empty!=6&empty!=7&empty!=8] up
       {board[empty]=board[empty+3]; board[empty+3]=0; empty=empty+3; c=c+2} -> Game();

Down() = [empty!=0&empty!=1&empty!=2] down
         {board[empty]=board[empty-3]; board[empty-3]=0; empty=empty-3} -> Game();

#define goal board[0] == 1 && board[1] == 2 && board[2] == 3 &&

#assert Game() reaches goal with min(c);
```
Strategy Analytics (MDP-based)
How can Federer beat Nadal?

- Federer vs Nadal is one of the greatest rivalries in tennis history.
- While Federer is widely regarded the best tennis player in history with 17 G.Slam (13 for Nadal) and 300+ weeks #1 (100+ for Nadal) and 6 ATP Year End Final Titles (0 for Nadal).
- However Nadal's record against Federer is 23-10 in Nadal's favour.
- Why? How can Federer beat Nadal

<table>
<thead>
<tr>
<th>Year</th>
<th>Australian Open</th>
<th>French Open</th>
<th>Wimbledon</th>
<th>US Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td>Andre Agassi</td>
<td>Juan Carlos Ferrero</td>
<td>Roger Federer</td>
<td>Andy Roddick</td>
</tr>
<tr>
<td>2004</td>
<td>Roger Federer</td>
<td>Gastón Gaudio</td>
<td>Roger Federer</td>
<td>Roger Federer</td>
</tr>
<tr>
<td>2005</td>
<td>Marat Safin</td>
<td>Rafael Nadal</td>
<td>Roger Federer</td>
<td>Roger Federer</td>
</tr>
<tr>
<td>2006</td>
<td>Roger Federer</td>
<td>Rafael Nadal</td>
<td>Roger Federer</td>
<td>Roger Federer</td>
</tr>
<tr>
<td>2007</td>
<td>Roger Federer</td>
<td>Rafael Nadal</td>
<td>Roger Federer</td>
<td>Roger Federer</td>
</tr>
<tr>
<td>2008</td>
<td>Novak Djokovic</td>
<td>Rafael Nadal</td>
<td>Rafael Nadal</td>
<td>Roger Federer</td>
</tr>
<tr>
<td>2009</td>
<td>Rafael Nadal</td>
<td>Roger Federer</td>
<td>Roger Federer</td>
<td>Juan Martin del Potro</td>
</tr>
<tr>
<td>2010</td>
<td>Roger Federer</td>
<td>Rafael Nadal</td>
<td>Rafael Nadal</td>
<td>Rafael Nadal</td>
</tr>
<tr>
<td>2011</td>
<td>Novak Djokovic</td>
<td>Rafael Nadal</td>
<td>Novak Djokovic</td>
<td>Novak Djokovic</td>
</tr>
<tr>
<td>2012</td>
<td>Novak Djokovic</td>
<td>Rafael Nadal</td>
<td>Roger Federer</td>
<td>Andy Murray</td>
</tr>
<tr>
<td>2013</td>
<td>Novak Djokovic</td>
<td>Rafael Nadal</td>
<td>Andy Murray</td>
<td>Rafael Nadal</td>
</tr>
</tbody>
</table>
enum {f_ad_ct, n_ad_ct, f_de_ct, n_de_ct};

// serve position: ad court or deuce court

enum {federer, nadal, na};

var turn = na; // serve turn;

var fscore = 0;
var nscore = 0;
var won = na;
var ball = 9;

WhoServe1st = []: {f_de_ct, n_de_ct}@ TossCoin{turn = i} -> Skip;

TieBreakGame = WhoServe1st;
(FedererServe [] NadalServe);
Probability distribution on Federer Serve

De_Fed1stServe = pcase { // all probability is based on percent %, 30 means 30%
   23: ServeT_in{ball= 6} -> NadForehandR // Nadal is lefty.
   15: ServeT_err{ball=9} -> De_Fed2ndServe
   30: ServeWide_in{ball =6} -> NadBackhandR
   11: ServeWide_err{ball=9} -> De_Fed2ndServe
   14: ServeBody_in{ball=6} -> (NadBackhandR [] NadForehandR)
   7: ServeBody_err{ball=9} -> De_Fed2ndServe};

De_Fed2ndServe = pcase { //1st serve is out
   15: ServeT_in{ball= 6} -> NadForehandR
   3: ServeT_err{ball=9} ->
   Fdoublefault{nscore++; if (nscore == 7) {won = nadal}
   else {if (turn == f_ad_ct){turn = f_de_ct} else {turn = n_ad_ct}}}
   33: ServeWide_in{ball =6} -> NadBackhandR
   2: ServeWide_err{ball=9} -> ...

   ...}

NextPt = FedererServe [] NadalServe [] (if won != na GameOver -> Skip);
Combine Real-Time and Probability [TOSEM’13]
(model checking C# program interface properties)

Passing me without stopping!
Given the C# Program of a lift algorithm

No need to understand the details of the code
#import "PAT.Lib.Lift";
#define NoOfFloors 3;
#define NoOfLifts 2;
var<LiftControl> ctrl = new LiftControl(NoOfFloors,NoOfLifts);
var passby = 0;

aSystem = (||| x:{0..NoOfLifts-1} @ Lift(x, 0, 1)) ||| Requests();
Requests() = Request();Request();
Request() = pcase {
  1 : extreq.0.1{ctrl.AssignExternalRequest(0,1)} -> Skip
  1 : intreq.0.0.1{ctrl.AddInternalRequest(0,0)} -> Skip
  1 : intreq.1.0.1{ctrl.AddInternalRequest(1,0)} -> Skip
  1 : extreq.1.0{ctrl.AssignExternalRequest(1,0)} -> Skip
  1 : extreq.1.1{ctrl.AssignExternalRequest(1,1)} -> Skip
...}
within[1];

Lift(i, level, direction) = case {
  ctrl.isToOpenDoor(i, level)==1: (serve.level.direction{ctrl.ClearRequests(i, level, direction)}) -> Lift(i, level, direction))
  ctrl.KeepMoving(i, level, direction)==1: (reach.level+direction.direction{passby =
    ctrl.UpdateLiftStatus(i, level, direction}) -> Lift(i, level+direction, direction))
  ctrl.HasAssignment(i)==1: changedirection.i{ctrl.ChangeDirection(i)} -> Lift(i, level, -1*direction)
  default : idle.i -> Lift(i, level, direction)
} within[2];
#define goal passby == 1;
#define assert aSystem reaches goal with prob;
To auto calculate:
- the maximum time delay of a critical event beyond which the overall system reliability will be compromised.
- the minimum probability shift of a specific event that will significantly tip the balance toward the winning strategy.
- …

Also look into:
- Linking to Operational Research and Machine Learning techniques/tools?
Some PAT info

- PAT is available at [http://pat.comp.nus.edu.sg](http://pat.comp.nus.edu.sg)
- 1 Million lines of C# code, 15 verification systems with 200+ build in examples, 100+ publications (CAV, FM, ICSE, ASE, TSE, TOSEM ...).
- Used as an educational tool in many universities.
- Attracted 3000+ registered users in the last 7 years from 800+ organizations in 72 countries, e.g. Microsoft, HP, Sony, Hitachi, Canon, Mitsubishi, NTT, Toyota ...
- Japanese PAT User group formed in Sep 2009:

  Founding Members:
  - Hiroshi Fujimoto
  - Nobukazu Yoshioka
  - Toshiyuki Fujikura
  - Kenji Taguchi
  - Masaru Nagaku
  - Kazuto MATSUI

  Commercialized in multiple countries, esp. in Japan, thanks to CATS!

Many Thanks to Ed!
- should have listened to you more seriously back in 1994
- Look forward to seeing you again soon in Singapore!