

## Propunere de proiect PBR

Învață regulile unui joc privind un joc în derulare. Exemple: table, joc de cărți...

Jocul de table (backgammon)

### Reguli

1. în joc sunt 2 jucători: red (R) și black (B);
2. toate piesele unui jucător merg în aceeași direcție, adică de la coloane cu numere mici către cele cu numere mari sau invers => NU depinde de secvență
3. la pornire se face o aranjare bine definită a pieselor în cele 4 cvartale => trebuie marcată starea inițială
4. jocul are 2 etape: în etapa I se aduc piesele în "casă"; în a II-a se scot din joc => NU depinde de secvență
5. fiecare zar dictează numărul de poziții în mutarea unei piese (exemplu: zarul 3-2 => mut o piesă 3 poziții și altă piesă, sau aceeași, - 2 poziții) => NU depinde de secvență
6. o dublă se mută de 2 ori => NU depinde de secvență
7. o piesă nu poate ajunge decât într-o coloană în care există deja piese de aceeași culoare, sau într-una goală sau într-una în care există exact o piesă a adversarului => NU depinde de secvență
8. o piesă singură într-o coloană poate fi scoasă din joc (în BAR) de adversar dacă piesa lui "cade" pe poziția piesei singure => NU depinde de secvență
9. o piesă scoasă din joc este repusă în cvartalul de plecare, cel mai îndepărtat de "casă" => NU depinde de secvență
10. regula "marțului": un jucător A este în situația de marț dacă adversarul a reușit să scoată toată piesele din casa lui înainte ca A să înceapă să scoată; se marchează ca 2 puncte => NU depinde de secvență, starea de marț este marcată
11. jucătorii sunt la mutare alternativ => depinde de secvență
12. începe jocul cel care dă mai mare dau dubla și dubla se mută => depinde de secvență

- alternativă: cifrele de pe zaruri s-ar fi putut suma, totalul lor dând numărul de poziții între care se schimbă piese pe tablă

### Strategii

- nu lăsa piese descoperite => pot fi scoase de adversar
- scoate mai întâi piesele aflate la distanță
- lasă în urmă câteva piese pentru a putea scoate adversarul
- ...

### Evenimente

- așezarea inițială a pieselor
- datul cu zarul: 1-1, ... 6-6 (o matrice simetrică 6 x 6)
- mutatul conform zarului
- scoaterea unei piese de culoare B când peste ea este plasată o piesă de culoare R (și invers)
- așezarea unei piese peste o alta de aceeași culoare
- toate evenimentele se desfășoară în secvență, deci reprezentarea trebuie să păstreze un indicator al timpului sau al numărului de ordine al evenimentului

## Reprezentare

- piesele aflate în joc

=> (checker <id> <player> <position: OFF/BAR/#col>)

- coloanele 1 - 24: o coloană reține toate piesele stivuite pe ea

=> (column <#column> <NULL/player> <idCh1>... <idChN>)

=> (bar <player> <id-ch1>... <id-chN>)

Un joc se poate afla într-una din fazele:

=> (phase <initial/play/final>)

- în *initial* piesele sunt aranjate pe tablă în poziția de start și jucătorii dar cu zarul pentru a afla cine mută primul, în *play* - se joacă o linie; în *final* - se actualizează punctajul.

În faza *play* un jucător se poate afla succesiv într-una din stările:

=> (state <player> <bring-home|take-off>)

- în *bring-home* jucătorul aduce piesele lui spre casă; când toate sunt în casă;

- în *take-off* - le scoate afară.

Când se află la mutare, un jucător execută următoarele secvențe, în ordine:

=> (sequence <player> <dice|think|move>)

- în *dice* el dă cu zarul;

- în *think* el gândește mutările conform zarului => se adaugă fapte (move NULL ...);

- în *move* el execută mutările gândite.

Secvența *think* interesează numai pentru B. R comunică direct mutările lui.

- un număr de fapte *dice*; unul singur poate avea valoarea NULL, ceea ce semnifică că zarul încă nu e jucat

=> (dice <time|NULL> <player> <x> <y>)

- mai multe fapte *move* satisfac un zar; dacă *time=*NULL, mutarea este doar planificată, fără să fi avut loc încă

=> (move <NULL/time> <player> <id-ch> <colSt:BAR/#col>  
<colDe:OFF/BAR/#col>)

=> (move <NULL/time> <player> NULL)

- rezultat: un fapt care să indice la sfârșitul unei linii care jucător a câștigat-o (cu detaliul numărului de puncte, pentru linie simplă sau marț); doar maximum 6 astfel de linii se găsesc simultan în bază

=> (line <player> <points>), unde <points> poate fi 0, 1 sau 2, 3

- la sfârșitul partidei, 3 puncte câștigate de același jucător se transformă într-o partidă câștigată

=> (game <player> <#won-games>)

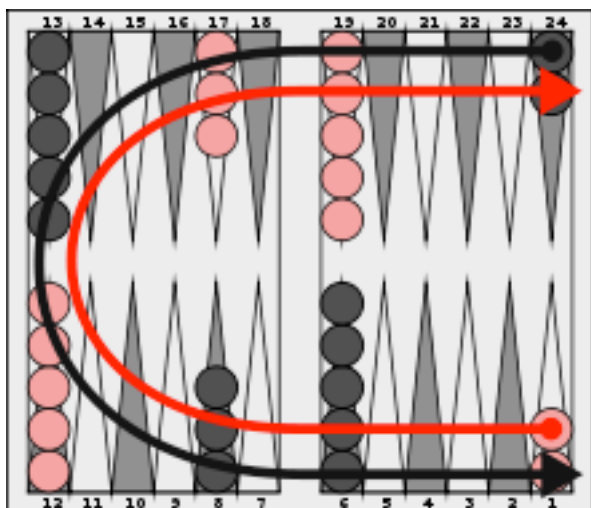


Figure 1: Paths of movement for red and black, with checkers in the starting position; viewed from the black side, with home or inner board at lower right (from <http://en.wikipedia.org/wiki/Backgammon>)

### Contextualități

Contextualities, affordances:

- mișcările sunt caracterizate prin direcții și sens.

Observații vizuale

- în jurul tablei: doi jucători

- tipuri de mișcări:

\* se dă cu zarul => în reprezentare apar elemente dice;

\* se mută piese: pe tablă, din tablă spre BAR, din BAR înapoi în tablă, din tablă în OFF => în reprezentare apar elemente move; argumentele lor colSt și colDe sunt corelate;

\* piesele se așează în stive în coloane => stiva este menținută în reprezentare;

\* când se mută, se ridică piesele aflate deasupra în stivă => scoaterea unei piese se face din poziția de vârf a stivei.

### Deducerea regulilor

- activitatea de deducere se desfășoară prin observarea tablei de joc, adică a secvenței de evenimente: dice, move;

- secvența totală de evenimente move descrie complet mișcările dintr-o partidă; din ea însă nu pot fi deduse regulile jocului pentru că lipsesc zarurile;

- regulile jocului ar trebui să poată fi deduse din secvențe contigue formate din dice urmat de move+, în timpii pentru move neaflând-se nici un alt dice;

- cum deduci că toate piesele unui jucător merg în aceeași direcție?

### Corelații

- tipul și direcția mișcării:

=> if colSt = BAR AND player = R then colDe < 7 are loc în 100% din cazuri, deși P= 25%: toate piesele sunt repuse în joc numai în primul sfert de tablă;

=> if colSt = BAR AND player = B, then colDe > 18 are loc în 100% din cazuri, deși P= 25%: toate piesele sunt repuse în joc numai în ultimul sfert de tablă;

=> **if colSt  $\neq$  BAR AND player = R then colDe-colSt > 0** în 100% din cazuri, deși P = 50%: toate mișcările lui R sunt dinspre valori mici către valori mari ale coloanelor;  
=> if colSt  $\neq$  BAR AND player = B then colDe-colSt < 0 în 100% din cazuri, P = 50%: toate mișcările lui B sunt dinspre valori mari către valori mici ale coloanelor.

### **Documentație**

Metode de învățare automată:

<http://ijcai.org/Past%20Proceedings/IJCAI-93-VOL2/PDF/022.pdf>

[http://deepblue.lib.umich.edu/bitstream/handle/2027.42/46972/10994\\_2004\\_Article\\_BF00994007.pdf?sequence=1&isAllowed=y](http://deepblue.lib.umich.edu/bitstream/handle/2027.42/46972/10994_2004_Article_BF00994007.pdf?sequence=1&isAllowed=y)

Metode de data mining:

Cartea care însoțește softul Weka: Witten&Frank, Data Mining, Practical Machine Learning Tools and Techniques, 2005 => [Index of /~pmihaela/books](#)  
Regulile de asociere: pag. 112-119 și puțin la pagina 419.