University of Ljubljana
Faculty of Computer and Information Science

Oleksandr Sivak

May a weak tennis player win?

graduate thesis

professional study programme of computer and information science

Mentor: izr. prof. dr. Gašper Fijavž

Ljubljana, 2015

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Oleksandr Sivak

Ne najmočnejši zmagovalci teniškega turnirja

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: izr. prof. dr. Gašper Fijavž

Ljubljana, 2015

Text is created by the word processor LaTeX.

The Faculty of Computer and Information Science issues the following thesis:

The paper *Fixing a Tournament* (Williams, AAAI 2010) exhibits a handful of sufficient conditions for a competitor to be a possible winner of a knock-out competition. Find the possible winners according to these conditions in the case of real data on tennis players. Find also the possible additional winners and finally make a comparison with randomly generated data.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Članek *Fixing a Tournament* (Williams, AAAI 2010) predstavi nekaj zadostnih pogojev, v skladu s katerimi lahko določimo zmagovalca turnirja na izpadanje. Poiščite možne zmagovalce v skladu z omenjenimi zadostnimi pogoji na primeru realnih podatkov o teniških igralcih. Poiščite tudi morebitne dodatne možne zmagovalce, na koncu pa naredite še primerjavo s slučajno generiranimi podatki.

Declaration of the diploma work authorship

Undersigned, Oleksandr Sivak, author of the graduate thesis with a topic:

May a weak tennis player win?

I confirm, that:

- the graduate thesis was made on my own, mentored by izr. prof. dr. Gašper Fijavž,
- the electronic copy, the title, the abstract of the graduate thesis are identical with the printed copy of the graduate thesis,
- I agree with publishing copies of the graduate thesis on the Internet through the university web archive.

Ljubljana, September 22, 2015 Signature of the author:

Contents

Abstract

The winner of a competition depends on the choice of actual matches played. We assume that each match is played between two players. Our goal is to examine which players can be made winners of a competition if we know any match result in advance. We only consider competitions in which the winner of a single match progresses to the next round and the loser leaves the competition. We focus on tennis competitions and use real data downloaded from atpworldtour.com. The final winner of a competition depends on the choice of matches in the first round — we call it a *bracket*. We would like to determine possible competition winners and for every winner π construct an appropriate *bracket* in which π is the winner. Apart from that we also study how tight are the sufficient conditions for a player to become a winner, as described in the paper *Fixing a Tournament* (Williams, AAAI 2010) [17]. For instance, one of our results is that a player whose relative rank is between 1 and 36 can with high probability be made a winner in a competition of 64 players.

Keywords: competition manipulation, fixing a tournament, tennis, fair deterministic winner, weak winners, weak players.

Povzetek

Zmagovalec tekmovanja je odvisen od začetnih pozicij igralcev. Omejimo se na primer, ko se v vsaki igri pomerita dva igralca. Naš cilj je ugotoviti, kateri igralci so lahko zmagovalci tekmovanja, če vnaprej poznamo vse možne rezultate dvobojev. Omejili se bomo na tekmovanja, kjer zmagovalec dvoboja napreduje v naslednji krog, poraženec pa je izločen iz tekmovanja. Osredotočili se bomo na teniške turnirje na podlagi realnih podatkov s spletne strani atpworldtour.com. Končni zmagovalec turnirja je odvisen od začetnih pozicij igralcev v prvem krogu — temu rečemo *razpored*. Določiti želimo vse možne zmagovalce tekmovanja in za vsakega zmagovalca π določiti ustrezen *razpored*. Poleg tega študiramo tudi, kako dobri so zadostni pogoji, ki jih opiše Williams v članku *Fixing a Tournament* (Williams, AAAI 2010) [17]. Kot primer, eden naših rezultatov pravi, da je lahko igralec, katerega relativna uvrstitev je med 1. in 36. mestom, z veliko verjetnostjo lahko zmagovalec teniškega tekmovanja s 64 udeleženci.

Ključne besede: manipulacija turnirja, tenis, deterministični zmagovalec, slabi zmagovalec, slabi igralec.

Razširjeni povzetek

V delu se ukvarjamo s problemom iskanja možnih zmagovalcev v uravnoteženih tekmovanjih na izpadanje. Teniški turnir je tipičen primer takšnega tekmovanja. Igralci v vsakem krogu odigrajo dvoboj, zmagovalci napredujejo v naslednji krog, poraženci so izločeni.

Tekmovanje na izpadanje lahko predstavimo z označenim dvojiškim drevesom. Koren drevesa je končni zmagovalec. Listi drevesa so udeleženi tekmovalci in za vsako notranje vozlišče velja, da je zmagalo v dvoboju, ki sta ga odigrala sinova. Oznaka notranjega vozlišča se torej ponovi v natančno enem sinu.

Zmagovalec turnirja je odvisen od

(i) izidov dvobojev med igralcema in

(ii) *razporeda.*

Denimo, da za množico igralcev ρ (vedno bomo prevzeli, da je število igralcev v ρ potenca števila dva) poznamo izide vseh možnih dvobojev. Želeli bi poiskati vse možne zmagovalce tekmovanja glede na razpored.

Naštejmo nekaj sorodnih rezultatov. Če ne zahtevamo, da je dvojiško drevo polno in uravnoteženo (v praksi to pomeni, da se lahko igralec priključi tekmovanju tudi v katerem kasnejših krogov in v najbolj enostavnem primeru postane zmagovalec z eno samo zmago v finalu), potem je problem iskanja vseh možnih zmagovalcev relativno enostaven. Lang, Pini, Rossi, Venable in Walsh [8] so dokazali, da lahko v množici n igralcev za izbranega igralca π v času $O(n^2)$ odločimo, ali obstaja dvojiško drevo (ne nujno uravnoteženo), pri katerem je π končni zmagovalec — in v primeru pozitivnega odgovora takšno drevo hkrati tudi zgradimo.

V primeru, ko izidi med pari igralcev niso deterministično določeni, temveč poznamo zgolj verjetnosti, da eden od igralcev zmaga, je problem drugačne narave. Problem določanja igralca, ki zmaga z največjo verjetnostjo, se imenuje problem najverjetnejšega zmagovalca. Pri izbranem igralcu π in realnem številu δ je NP-težko odločiti, ali obstaja razpored, pri katerem je z verjetnostjo vsaj δ igralec π končni zmagovalec turnirja [14].

V primeru polnega in uravnoteženega dvojiškega drevesa, s katerim lahko predstavimo tekmovanje, in determinističnih izidov, ko za vsak par igralcev poznamo izid njunega medsebojnega dvoboja, je računska težavnost iskanja vseh zmagovalcev odprt problem. Williams je v članku *Fixing a tournament* [17] opisala tri zadostne pogoje, pri katerih je igralec tudi zmagovalec takšnega tekmovanja.

Če je ρ množica igralcev in R množica izidov (matrika, ki za vsak par igralcev π, π' določi rezultat njunega dvoboja) potem z $\text{out}(\pi)$ označimo število igralcev, ki jih π premaga, in z $\text{in}(\pi)$ število igralcev, ki premagajo igralca π. Williams [17] je pokazala, da če velja za igralca π eden od pogojev

(STR) za vsakega igralca π', ki premaga π, mora veljati $\text{out}(\pi) \geq \text{out}(\pi')$ ali

(KNG) $\text{out}(\pi) \geq |\rho|/2$ in za vsakega igralca π', ki premaga π, obstaja igralec π'', ki premaga π' in izgubi proti π, ali

(SKG) za vsakega igralca π', ki premaga π, obstaja vsaj $\log_2(|\rho|)$ igralcev, ki premagajo π' in izgubijo proti π,

potem je π možen zmagovalec tekmovanja. V tem delu pokažemo tudi pravilnost pogojev (STR), (KNG) in (SKG).

S spletne strani ATP [1] smo naložili dejanske podatke o teniških igralcih in njihovih medsebojnih dvobojih. Rezultat vsakega para smo ovrednotili glede na historične podatke njunih dvobojev. Lotili smo se naslednjih vprašanj.

Kako slab igralec na tipičnem teniškem tekmovanju je lahko zmagovalec glede na enega od pogojev (STR), (KNG) in (SKG)? Pri tem kvaliteto igralca merimo bodisi z uvrstitvijo na ATP lestvici ali pa z relativno uspešnostjo v družini vseh udeležencev tekmovanja. Pri testiranju smo izločili nekaj najboljših igralcev. Za njih domnevamo, da vedno pripadajo možnim zmagovalcem. Po 10.000 ponovljenih poskusih lahko trdimo, da manager povprečnega ATP igralca lahko prevzame, da njegov igralec bo zmagal v tekmovanju, če je razvrščen med 1 in 36 igralci bodisi na ATP ali pa na lestvici z relativno uspešnostjo v družini vseh udeležencev tekmovanja.

Kako dobri so pogoji (STR), (KNG) in (SKG)? Ali lahko v realnosti pričakujemo tudi zmagovalca, ki ne ustreza nobenemu od pogojev (STR), (KNG) in (SKG)? Z relativno enostavno konstrukcijo tekmovanja z 8 igralci smo uspeli poiskati zmagovalca tekmovanja, ki ne zadošča niti (STR), (KNG) niti (SKG).

Lahko takšen fenomen pričakujemo tudi v realnosti? Izvedli smo 10.000 testov na slučajno izbranih 64 igralcih izbranih iz večjega nabora 148 igralcev. Zdi se, da vsak zmagovalec teniškega tekmovanja, če upoštevamo realne podatke, tudi izpolnjuje enega od pogojev (STR), (KNG) ali (SKG).

Poskus smo nadaljevali s slučajno generiranimi podatki. Pričakovano smo eksperimentalno potrdili dejstvo, da pri slučajno generiranih podatkih vsi igralci lahko postanejo zmagovalci tekmovanja. Če z n označimo število igralcev, potem za slučajno izbranega igralca π pričakujemo približno $n/2$ igralcev, ki premagajo igralca π. Za vsak par igralcev π, π' pa pričakujemo približno $n/4$ igralcev, ki izgubijo v dvoboju s π in premagajo π'. Skratka, vsak posamezen igralec π z veliko verjetnostjo zadošča pogoju (SKG) in je posledično možen zmagovalec tekmovanja.

Za testiranje smo izdelali programsko opremo za naslednje naloge:

- zajem in interpretacija podatkov o igralcih in dvobojih z spletnega portala atpworldtour.com,
- določanje možnih zmagovalcev glede na pogoje (STR), (KNG) ali (SKG), pri izbrani množici igralcev in predpisanih izidih,

- v primeru, da igralec π zadošča kateremu od pogojev (STR), (KNG) ali (SKG), tudi izračun ustreznega razporeda, pri katerem π postane zmagovalec,
- slučajno generiranje razporedov tekmovalcev in posledično izračun zmagovalca pri izbranem razporedu,
- posredovanje izmerjenih podatkov na spletno stran za izdelavo diagramov plot.ly.

Introduction

In this world it is natural to compete. We need to define a person who is a winner in a given branch of sport, science, business... In this work we focus on tennis.

Given a set of players, how to choose a winner? One of the most popular formats is a single-elimination competition, also called a knockout competition. For instance, Wimbledon, Roland Garros use the knockout format, because it nicely defines a winner. But, the result is not stable in the knockout format. The knockout competition proceeds in rounds. In each round players are paired up to play a game, then the round winners move on to the next round, whereas the losers leave the competition. The result of a knockout competition depends on the first round arrangement of all players in the given set. We call such an arrangement a *bracket*.

Novak Djokovic is at the time of writing considered one of the best tennis players in the world. It should not be difficult to pair players in the first round so that Djokovic emerges as the winner of the competition. What about the players that lie much lower in the ATP ranking? Can they also be made winners if we are allowed to rearrange the pairs?

Williams [17] describes several conditions under which a player is a winner in a knockout competition if the *bracket* is made in his/her favor. We tested real data looking at which players can become winners, according to each of the conditions.

We downloaded the appropriate data from the official ATP (Association of professional tennis players) web-resource atpworldtour.com, which lists players and matches played.

A winner in a pair of players of our competition is the one who has more wins in matches between them. In case of a tie we break ties according to the sum of prize money earned during their careers.

We used two ranking systems, one is based on ATP, the other on a number of opponents a player beats — we call it the *outdegree* system.

Each knockout competition starts with a number of players which is a power of two. Usually 128 players for largest competitions. We wrote code to download the appropriate data, and also to parse and analyze it. Later, a testing environment was established and several probabilistic tests were executed.

In September 2015 we downloaded data of the first 148 players visible on the ATP web-resource and scores were defined only between those players. We tested sets of 64 players, chosen from the set of 148 players excluding the first top 20 players, in the first round. For such a set of players we determined which ones can be winners according to one of Williams' conditions [17]. Also, we made 10.000 permutations of a bracket with 64 players without the first 20 top players to compute real winners and compare them with winners according to one of the conditions. Finally we generated 100 random scores for 128 players to verify that real data is not random.

On average, players with ranks between 1 and 36, both by ATP and outdegree, can be winners.

Next, the conditions of Williams are good enough in the sense that they included all the real winners computed among 10.000 brackets.

Finally, in the random world every player satisfies one of the the conditions. Also, almost every player π is a *super-king* as for every player π' winning over π there are more than $\log_2(n)$ players who lose to π and beat π'.

This work is a practical example for those interested in research on how to make a player a winner, even if this player is not the strongest one.

Chapter 1

Basics

In this chapter we exhibit several representations of a *knockout competition* and develop the necessary notation. We finish the chapter by discussing a probabilistic approach to define results in matches for each pair of players.

1.1 A very small competition

Generally, a *knockout competition* includes a collection of players ρ and, in our work, a collection of match results R — given a pair of players, their match result is determined. We assume the number of players in ρ to be a power of 2, and we define R as a mapping from $\rho \times \rho$ into $\{1, 0, \mathrm{X}\}$.

We consider, as an introductory example, a very small *knockout competition* with a collection of players $\rho_0 = \{\text{Wawrinka, Nishikori, Murray, Berdych}\}$. Given that R_{ρ_0} is a mapping from $\rho_0 \times \rho_0$ into $\{1, 0, \mathrm{X}\}$, we can represent R_{ρ_0} in a tabular way as a matrix.

Let us fix an ordering of players in ρ_0 as (Wawrinka, Nishikori, Murray, Berdych). The *outcome matrix* of R_{ρ_0}, with respect to the chosen ordering, is a matrix $R_{\rho_0} = (r_{i,j})_{i,j}$, with entries in $\{1, 0, \mathrm{X}\}$, so that $r_{i,j}$ indicates who among players i and j is a winner in a match. See Figure 1.1.

	Wawrinka	Nishikori	Murray	Berdych
Wawrinka	X	1	0	1
Nishikori	0	X	0	1
Murray	1	1	X	0
Berdych	0	0	1	X

Table 1.1: Matrix R_{ρ_0} showing results of matches between players from ρ_0.

Every diagonal entry $r_{i,i}$ is equal to X and every off diagonal entry $r_{i,j}$ is either 0 or 1 so that if $r_{i,j} = 0$ then $r_{j,i} = 1$. For instance, consider the entry $r_{1,2}$ which equals to 1: this indicates a win of Wawrinka over Nishikori. The entry $r_{2,1}$ is 0, which indicates that Nishikori loses to Wawrinka .

On the other hand, we can represent R_{ρ_0} also as a *tournament* — a complete oriented graph R_{ρ_0} so that there exists an arrow from a player π to a player π' if and only if π beats a player π'. See Figure 1.1.

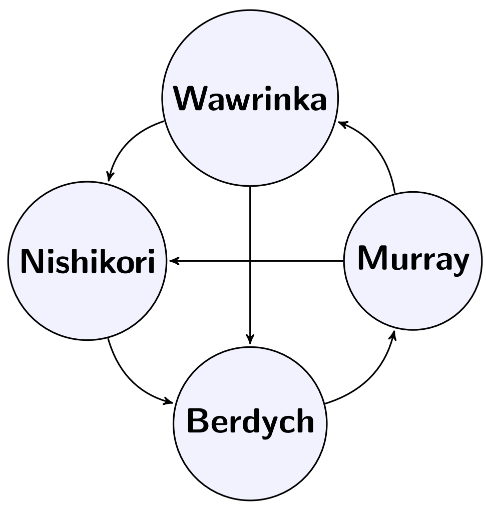

Figure 1.1: Tournament graph R_{ρ_0} showing results of matches between players from ρ_0.

A collection of match results R includes all players taken from ρ. A number of players in ρ may vary. Which representation of match results should be used, and when? We would use a graph when we can grasp all players in the graph at a glance — the number of players in ρ should be either from 8, 4, and 2. On the other hand, given a pair of players, in a matrix R we match a row with a column to find an entry — this way should be preferred, when the number of players is greater than 8.

Let us apply a collection of players ρ_0 and a collection of results R_{ρ_0} to construct the first instance of a *knockout competition*.

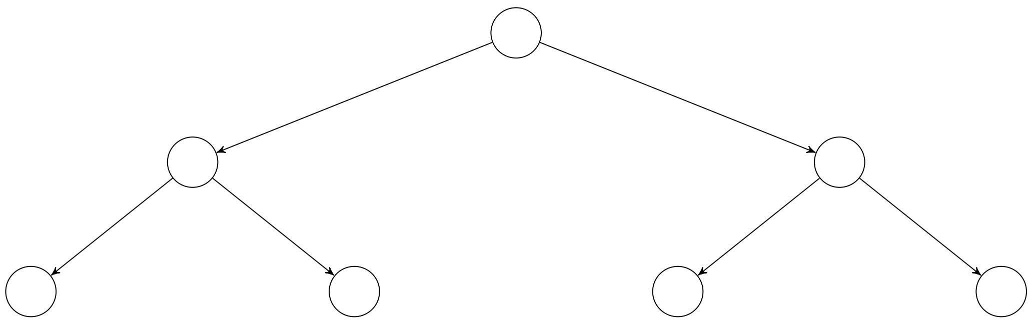

Figure 1.2: A *binary tree* T.

The *complete binary tree*, depicted in Figure 1.2, is a skeleton for our *knockout competition*. We intentionally used 4 leaves in the *binary tree*, as this is exactly the number of players in ρ_0. Let us make a *bijective mapping* from ρ_0 into leaves of the *binary tree* so that each leaf is labeled a player, and denote the mapping by μ_0. See Figure 1.3.

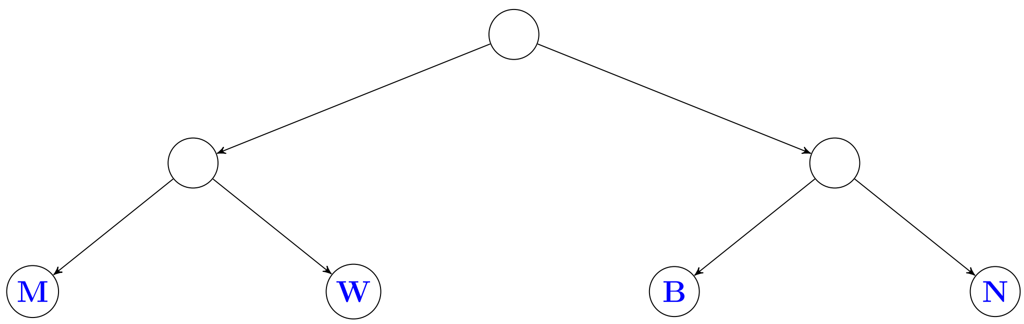

Figure 1.3: A *binary tree* including μ_0.

In Figure 1.3 and subsequent figures, M, W, B, N stand for Murray, Wawrinka, Berdych, Nishikori, for the sake of smaller figures. Pairs of leaves (M, W) and (B, N) have one and common non-leaf node adjacent to them. Effectively, each non-leaf node represents a match between its children, and is labeled a winner with respect to R_{ρ_0}.

Let us assume that there is another mapping made by exchanging positions of leaves with players Murray and Wawrinka in μ_0. But, our competition matches will be still the same — Murray plays against Wawrinka, and Nishikori — against Berdych. We state that both mappings are equivalent:

Definition 1.1 *A* bracket *is an equivalence class of bijective mappings from ρ into leaves of a complete binary tree. Two such mappings are equivalent if the same matches take place at all rounds of a competition with respect to R.*

We denote the *bracket* by β_0 and state:

Definition 1.2 *A* complete binary tree of a bracket*, or a* competition tree*, is a labeling of nodes in a complete binary tree expanding a bracket so that each leaf is labeled with respect to a mapping from the bracket, and each non-leaf node labeled ω has exactly two children ω and ω', where ω beats ω' with respect to a collection of match results R.*

The *competition tree* of β_0 is denoted by T_{β_0}, and is depicted in Figure 1.4.

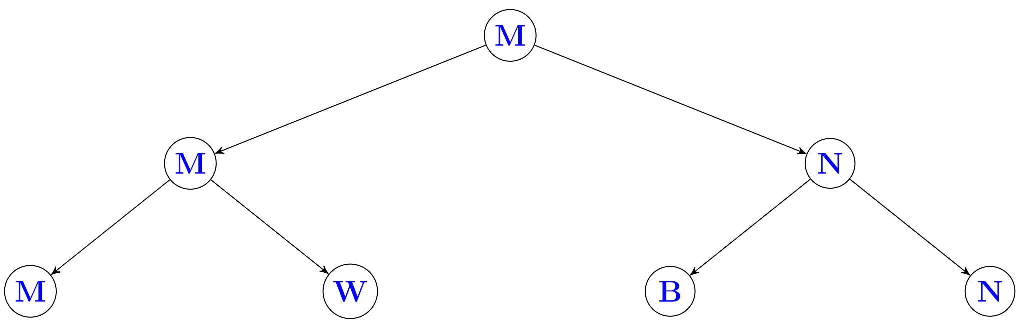

Figure 1.4: T_{β_0} with respect to R_{ρ_0}.

We have chosen a *complete binary tree* to represent a *knockout competition*, because a *complete binary tree* visually consists of levels going from the bottom to the top, whereas each level includes a number of nodes of a power of 2. Also, starting from the second level a number of nodes is halved with respect to the lower level. Effectively, a *complete binary tree* follows a *knockout competition* — each level, except the top one, represents a round in a *knockout competition*.

Definition 1.3 *A* winner *is a root of a competition tree.*

Let us observe that the ordering of leaves in T_{β_0}, depicted in Figure 1.4, is in no relation to the ordering used to label rows and columns of R_{ρ_0}. Nevertheless, the ordering of leaves determines a winner in our competition if results of distinct matches are defined by R_{ρ_0}. An alternative competition with a different winner, this time Wawrinka, is shown in Figure 1.5.

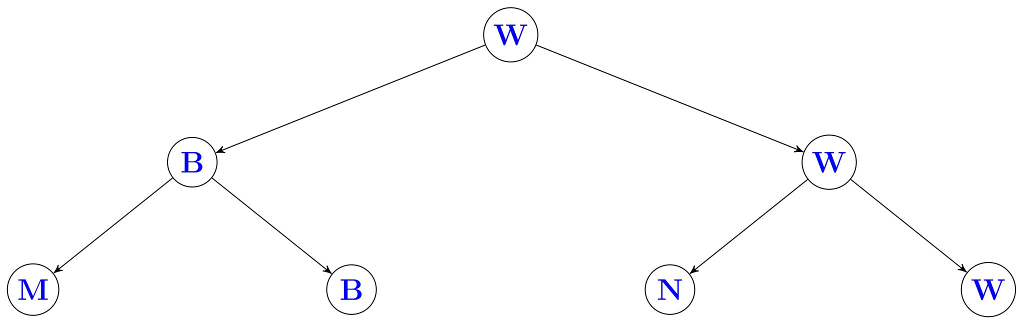

Figure 1.5: T_{β_1} with respect to R_{ρ_0}.

One of the ways to construct different competitions tree is to perturb starting positions of players — this is called *fixing a tournament* [17]. Starting positions of players relate to a *bracket*, and *fixing a tournament* is also called *rigging a bracket* [11]. We have *fixed a tournament* R_{ρ_0} so that Murray is a *winner* in T_{β_0} and Wawrinka in T_{β_1}. But, we cannot *rig a bracket* for either Nishikori, or Berdych, because each of them beats 1 player in ρ_0 which includes 4 players. We have essentially determined all winners in the introductory example — this the very gist of our work. But, we are more interested to *rig a bracket* for players not at the top, like Berdych or Nishikori.

In practice, a tennis competition includes 128 players. We need another representation of a *knockout competition* for the sake of convenience.

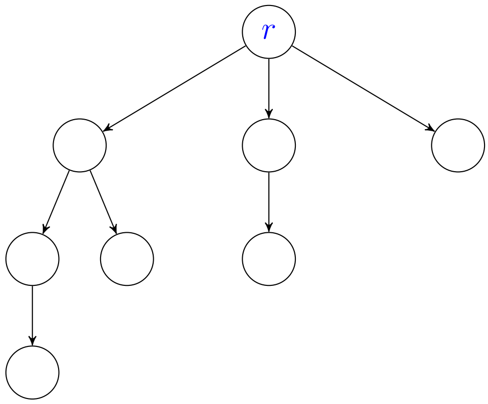

Figure 1.6: A binomial tree B.

Let us consider a tree B, depicted in Figure 1.6. The tree root, r, has three children or subtrees, denoted by B_1, B_2 and B_3. Let us also assume that the indices are chosen in the increasing order according to the number of vertices in the subtrees. Clearly, these numbers are powers of two, $1 = 2^0$, $2 = 2^1$ and $4 = 2^2$. The union of B_1 and r is *isomorphic* to B_2, and the union of B_1, B_2 and r is *isomorphic* to B_3. We call the tree B in Figure 1.6 a *binomial tree*. We can define *binomial trees* in several different ways.

Definition 1.4 *A* binomial tree B *of order* 2^n, *where* $n \geq 0$, *is a rooted tree with a root* r *so that*

1. *if* $n = 0$, *then* B *is a singleton-root tree having a single vertex, and*

2. *if* $n \geq 1$, *then* r *has a child* r', *for which the subtree* B' *of* B *rooted at* r' *is a* binomial tree *of order* 2^{n-1}, *and also* $B - B'$ *is a* binomial tree, *rooted at* r, *of order* 2^{n-1}.

Given a *binary tree of a bracket* T_β we can construct a *binomial tree of a bracket* B_β by contracting every edge whose endvertices both have the same labels, and taking the vertex of maximal degree as a root. For instance, in T_{β_1}, depicted in Figure 1.5, Wawrinka is repeated 3 times and Berdych twice. Let us contract edges whose endvertices have Wawrinka or Berdych, and take Wawrinka as a root of B_{β_1}. See Figure 1.7.

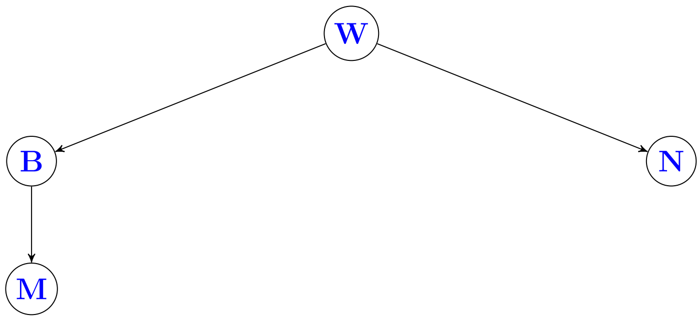

Figure 1.7: B_{β_1} with respect to R_{ρ_0}.

Essentially the *binomial tree of a bracket*, B_{β_1}, depicted in Figure 1.7, and the *binary tree of a bracket*, T_{β_1}, depicted in Figure 1.5, carry the same information.

Claim 1.1 *Every* binomial tree of a bracket B_β *uniquely corresponds to a* binary tree of a bracket T_β.

Vice-versa, the following function *binaryFromBinomial* takes a *binomial tree of a bracket* B_β and the tree root r as an input, and returns the *complete binary tree of a bracket* as the output.

function BINARYFROMBINOMIAL(B_0, r)
    if COUNT(vertices in B_0) $= 1$ **then return** B_0
    else
        $r' \leftarrow$ a son of r so that the subtree rooted at r' is maximal
        $B' \leftarrow$ the tree rooted at r'
        lsubtree $\leftarrow$ BINARYFROMBINOMIAL(B', r')
        rsubtree $\leftarrow$ BINARYFROMBINOMIAL($B - B'$, r)
        return T_β rooted at r whose subtrees are *lsubtree* and *rsubtree*
    end if

```
10: end function
```

Let us follow the lines of the function *binaryFromBinomial*, given the *binomial tree of a bracket* B_{β_1}, depicted in Figure 1.7, and the root Wawrinka as the input. We jump to the line 4, because B_{β_1} includes 4 vertices. The vertex with Berdych is assigned r' and at the line 5 the two nodes *binomial tree* with Berdych and Murray is assigned B'. But, the function is called recursively at the line 6, this time with B' and r' as the input.

Let us consider the function call, again. We jump to the line 4, because B_0 includes 2 vertices, Berdych and Murray. The only vertex rooted at r, Murray, is assigned r', and the singleton-root *binomial tree*, rooted at r', is assigned B'. The next call of *binaryFromBinomial* at the line 6, with B' and r' as the input, executes the line 2 so that the returned *binary tree* with only one node, Murray, is assigned $lsubtree$. But then, at the line 7, the call of *binaryFromBinomial*, this time with the *binomial tree* rooted at Berdych and the tree root as the input, executes the line 2 so that the returned *binary tree* with only one node, Berdych, is assigned $rsubtree$. Effectively, the *binary tree of a bracket*, rooted at r, Berdych, with children $lsubtree$ and $rsubtree$ is returned at the line 8.

Let us return to the very first call of *binaryFromBinomial*, and we have $lsubtree$ assigned to the *binary tree of a bracket* with *winner* Berdych. Analogously, the function call at the line 7, this time with the *binomial tree* including Wawrinka and Nishikori, and the tree root, Wawrinka, as the input, returns the *binary tree of a bracket* rooted at Wawrinka and is assigned $rsubtree$. Finally, at the line 8, we construct the *binary tree of a bracket* T_β rooted at r, Wawrinka, so that the two subtrees, $lsubtree$ and $rsubtree$, are children of r.

1.2 Notation

In this section we introduce the necessary notation and definitions. Given a collection of players ρ, a collection of match results R, and a player $\pi \in \rho$, the set of players who lose to π is denoted by

$$N^{\text{out}}(\pi) \;=\; \{\pi' \in \rho,\; r_{\pi,\pi'} \;=\; 1\} \tag{1.1}$$

This is exactly the set of outneighbours of π in a *tournament* R. The outneighbours of Wawrinka, with respect to the *tournament* R_{ρ_0}, depicted in Figure 1.1, are N^{out}(Wawrinka) = {Nishikori, Berdych}. See Figure 1.8.

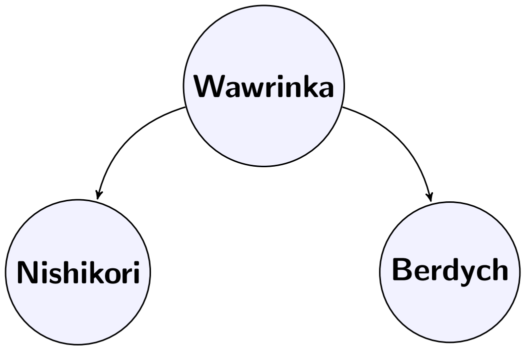

Figure 1.8: Outneighbours of Wawrinka — Nishikori and Berdych.

Further, for a subset of players χ, $\chi \subseteq \rho$, we set

$$N^{\text{out}}_{\chi}(\pi) \;=\; N^{\text{out}}(\pi) \cap \chi. \tag{1.2}$$

Analogously,

$$N^{\text{in}}(\pi) \;=\; \{\pi',\; r_{\pi,\pi'} \;=\; 0\} \tag{1.3}$$

is the set of players who beat a player π, and is called the set of inneighbours of π. For instance, with respect to R_{ρ_0}, the inneighbours of Nishikori are Wawrinka and Murray, N^{in}(Nishikori) = {Wawrinka, Murray} — this is depicted in Figure 1.9.

Again, for a subset $\chi \subseteq \rho$ we set

$$N^{\text{in}}_{\chi}(\pi) \;=\; N^{\text{in}}(\pi) \cap \chi. \tag{1.4}$$

Let

$$\text{out}\,(\pi), \tag{1.5}$$

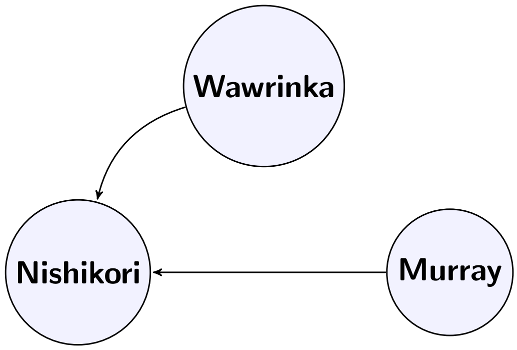

Figure 1.9: Inneighbours of Nishikori — Wawrinka and Murray.

$$\mathrm{out}_{\chi}(\pi), \tag{1.6}$$

$$\mathrm{in}\,(\pi), \tag{1.7}$$

$$\mathrm{in}_{\chi}(\pi) \tag{1.8}$$

denote the cardinalities of $N^{\mathrm{out}}(\pi)$, $N^{\mathrm{out}}_{\chi}(\pi)$, $N^{\mathrm{in}}(\pi)$, $N^{\mathrm{in}}_{\chi}(\pi)$, respectively. We call $\mathrm{out}\,(\pi)$ and $\mathrm{in}\,(\pi)$ the *outdegree* and the *indegree* of π.

Consider R_{ρ_0} in Figure 1.1, our exemplary collection of match results. Berdych is the only player who beats Murray. But Nishikori beats Berdych and loses to Murray. For Murray and for every player π who beats Murray there is a player π' so that Murray beats π' and π' beats π. This also holds for Wawrinka, because he has Berdych as an outneighbour to beat Murray.

Definition 1.5 *A* king *is a player $\pi \in \rho$, so that for every player $\pi' \in N^{in}(\pi)$ we have $in_{N^{out}(\pi)}(\pi') = \left|N^{in}_{N^{out}(\pi)}(\pi')\right| \geq 1$. In other words, for every player π' that beats π there is a player π'' so that π beats π'' and π'' beats π'.*

Also, notice that Murray and Wawrinka have only one inneighbour in our competition. Murray is beaten by Berdych and Wawrinka by Murray.

Is there a difference between Murray and Wawrinka? Notice that Murray has two outneighbours who beat Berdych in our competition, Wawrinka and Nishikori. This does not hold for Wawrinka. Namely, Berdych is the only player who beats Murray, a winner over Wawrinka.

Definition 1.6 *A player π is a* super-king *if for every player $\pi' \in N^{in}(\pi)$ we have $in_{N^{out}(\pi)}(\pi') \geq \log_2(|\rho|)$. Or equivalently, for every player π' that beats π there is at least $\log_2(|\rho|)$ outneighbours of π, and all of them beat π'.*

Note that every *super-king* is also a *king*. But, a *king* may not be a *super-king*. For instance, in our collection ρ_0 Wawrinka is a *king*, but not a *super-king*, with respect to a collection of match results R_{ρ_0}. Murray, on the other hand, is both a *king* and a *super-king*. *Kings* and *super-kings* were first defined in [11].

1.3 Probabilistic approach

Given a collection of players ρ, a collection of match results R, and a player $\pi \in \rho$, how difficult is it to *fix a tournament* R so that π becomes a winner?

We consider R so that, given a match between two players, say π and π' from ρ, the match result is known. But, a match result may be defined either deterministically, with π either losing or winning the match, or probabilistically, when the outcome maybe defined by a probability of π winning the match. A competition may be represented, with respect to R, either by a *general binary tree*, when few players from ρ may not play in the first round, or by a *complete binary tree*, when every player from ρ is included in the first round.

Let us consider the deterministic approach for a *general binary tree*. In section 1.1 we could not *fix a tournament* R_{ρ_0}, depicted in Figure 1.1, for Nishikori or Berdych, because they could win only one round in a two rounds competition. We can construct an *unbalanced complete binary tree of a bracket* β_0 so that Berdych is a *winner* in this tree. This is depicted in Figure 1.10.

In the first round of our competition Murray plays against Nishikori, but Wawrinka plays against Murray in the second round so that Murray proceeds to the final to play against the only inneighbour, with respect to the *tournament* R_{ρ_0}, Berdych. There are only two players from ρ_0, Murray

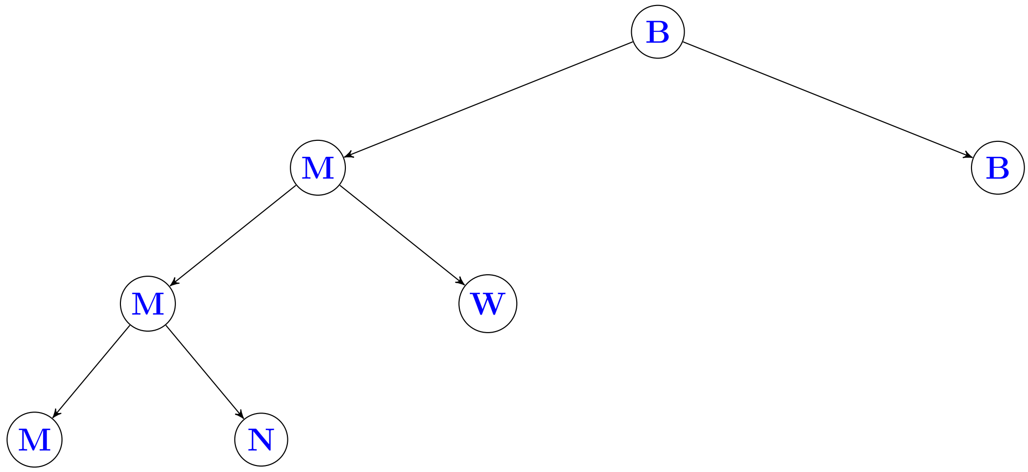

Figure 1.10: An unbalanced T_{β_0} with respect to R_{ρ_0}.

and Nishikori, who play in the first round of our competition. As shown, it is possible to *rig a bracket* for Berdych in an *unbalanced complete binary tree*, depicted in Figure 1.10. Lang et al. [8] state that it is efficient for a chair to *fix a tournament* in *general binary trees*, which are also called *unfair* in [8]. Namely, given n players in ρ, Vu et al. [14] showed that the time complexity to find a *bracket*, that is not necessarily balanced, for π so that π becomes a winner is $O(n^2)$, see also [8].

Fixing a tournament for the deterministic approach in a *balanced complete binary tree* is the study topic of Williams in [17]. She found several sufficient conditions for a player to be a winner — we discuss them in the next chapter. It is currently not known how difficult it is to compute all winners for *balanced complete binary trees*. But, with the assumption that some players cannot meet in a match, it was shown in [17] that the problem is NP-hard. Note that Stanton and Williams [11] refer to the *tournament fixing problem* only when match results are either 0 or 1, namely, only for the deterministic approach.

What if a winner of a pair of players was defined probabilistically? Vu et al. discussed this topic in [14]. Given a real number δ, they showed that it is NP-hard to *fix a tournament* in a *balanced complete binary tree* so that the probability of π winning a *knockout competition* is at least δ. Analogously, it is hard to determine the maximal probability for π to become a winner. Further, it still remains an open question whether there exists an efficient algorithm to find all winners in an *unbalanced complete binary tree*. Vu et al. [14] state that a tree structure should be biased towards a target player, an expected winner. We depicted such tree structure in Figure 1.10 for Berdych so that he plays one game against a player he can beat, in the final of our competition.

We have in our work focused solely on *deterministic fair winners*. We realize that this deterministic approach might not be the most realistic one.

Chapter 2

Fair deterministic winners

In this chapter we expose and prove heuristics, which were first proved by Williams [17], to find as many fair deterministic winners as possible.

2.1 The necessary condition

Assume that we can partition a collection of players ρ into a pair of subsets ρ_ω and ρ_λ so that every player from ρ_ω beats all players in ρ_λ. We also assume that ρ_ω and ρ_λ are not empty. Evidently, no player from ρ_λ can become a winner with respect to R. Vice versa, if π is a winner in our competition, then $\pi \in \rho_\omega$. But, we need to knock-out players from $\rho_\omega \setminus \{\pi\}$.

In particular, if ρ_ω is a single-player subset, then its element, player $\pi \in \rho_\omega$, is called the *Condorcet winner* and he wins in every possible *bracket*. We will also implicitly exclude Condorcet winners. We will first consider the *outdegree* of π in ρ_ω to find a sufficient condition for π to become a winner.

2.2 The sufficient conditions

Let $\pi \in \rho$ and R be given. We say that π satisfies *nodes stronger than the nodes that beat them*, or (STR) condition, if for every player $\pi' \in N^{\text{in}}(\pi)$ we have $\text{out}(\pi) \geq \text{out}(\pi')$. Next, we say that a player π is a *king who beats* *half the players*, or satisfies (KNG) condition, if $\text{out}(\pi) \geq |\rho|/2$ and for every player $\pi' \in N^{\text{in}}(\pi)$ there exists a player $\pi'' \in N^{\text{out}}(\pi)$ so that π'' beats π'. Finally, we say that π is a *super-king*, or satisfies (SKG) condition, if for every player $\pi' \in N^{\text{in}}(\pi)$ there exist at least $\log_2(|\rho|)$ players from $N^{\text{out}}(\pi)$ who beat π'.

Now, Williams [17] has proven that (STR), (KNG) and (SKG) are sufficient conditions to become a winner.

Theorem 2.1 *Let ρ be a collection of players, $\pi \in \rho$, and R a collection of results. If π satisfies (STR) or (KNG) or (SKG), then π is a possible winner.*

If a player satisfies a sufficient condition (STR) or (KNG) or (SKG), then we call such a player a *W-winner*. We will prove Theorem 2.1 in three separate cases in the next three sections. We will in every case show how to construct a *competition tree* with π emerging as the *winner*.

2.3 Strong players

In order to prove the (STR) case of Theorem 2.1, let us fix a player π, and assume that he matches condition (STR).

Let us first observe the very small cases. If $|\rho| = 1$ then π is the only player in ρ and no matches are needed. If $|\rho| = 2 = 2^1$, then the only remaining player π' clearly loses in a match with π. Namely, if π' would have won over π, then $\pi' \in N^{\text{in}}(\pi)$, and (STR) implies that $1 = \text{out}(\pi') \leq \text{out}(\pi) = 0$, which is clearly a contradiction.

Having treated the small and easy cases, let us turn our attention to bigger collections of players. We will henceforth assume that $|\rho| = 2^k \geq 4$, or equivalently $k \geq 2$. We will also assume that π is not the Condorcet winner.

We shall construct a bracket for which π emerges as the winner using binomial trees. Let $A = N^{\text{out}}(\pi)$ be the set of players who get beaten by π, and let $B = N^{\text{in}}(\pi)$ be the set of players which beat π. Now (A,B) is a partition of $\rho \setminus \{\pi\}$, where both A and B are nonempty sets of players. The set B is nonempty as by assumption π is not the Condorcet winner, and given an arbitrary $\pi' \in B$ we have (by (STR)) also $|A| = \text{out}(\pi) \geq \text{out}(\pi') \geq 1$ implying that also A is nonempty.

(1) Assume that the players from $\rho \setminus \{\pi\}$ can be partitioned into *binomial trees of brackets* whose roots lie in A. Then π is a winner.

Let $B_0, \ldots, B_r$ are binomial trees of brackets whose respective roots $a_1, \ldots, a_k$ all lie in A. Assume that for a pair of different indices i, j binomial trees B_i and B_j are of the same order 2^ℓ. If, say, π_i beats π_j, then B_i and B_j can be joined into a single binomial tree of order $2^{\ell+1}$ with root π_i. Recall that the root π_i lies in A. By repeated joining of binomial trees of the same order we can assume that in the (final) sequence of binomial trees $B_0, \ldots, B_r$ no pair of trees have the same order.

Now, trees $B_0, \ldots, B_r$ form a partition of (all) vertices of $\rho \setminus \{\pi\}$, the total number of players in their (disjoint) union is equal to $2^k - 1$, and no two of them are of the same order. As the only way to rewrite a number $2^k - 1$ as a sum of different powers of 2 is

$$2^k - 1 = 2^0 + 2^1 + \cdots + 2^{k-1}$$

we may without loss of generality (by permuting the indices if necessary) assume that $r = k - 1$ and for every $i \in \{0, \ldots, k-1\}$ we have exactly 2^i players in the binomial tree of a bracket B_i. Recall that the root of B_i is $\pi_i \in A$, and gets beaten by π in a match.

Now let us construct a binomial tree B by making π its root with all $B_0, \ldots, B_{k-1}$ as its subtrees. As π beats all $\pi_0, \ldots, \pi_{k-1}$ and the number of nodes in the tree is $|\rho|$ we have constructed a binomial tree of a bracket with π as its root. Hence π is a winner and the proof of (1) is complete.

Our next problem is partitioning the players of $\rho \setminus \{\pi\}$ into *binomial trees of brackets*. Let us first give an intermediate argument.

(2) Given π let A and B be its out- and inneighbours, respectively. If π satisfies (STR), then for every $\pi' \in B$ we have $\text{out}_B(\pi') < \text{in}_A(\pi')$. In

other words, every player $\pi' \in B$ gets beaten by strictly more players from A compared to the number of players π' beats in B.

This can be shown by manipulating the inequalities concerning degrees of players. Choose an arbitrary player $\pi' \in B$. By (STR) we have

$$\text{out}\,(\pi') \leq \text{out}\,(\pi) = |A|.$$

Now π' on one hand beats π, and on the other π' may also beat some players from A or B. Hence

$$\text{out}\,(\pi') = 1 + \text{out}\,_A(\pi') + \text{out}\,_B(\pi').$$

Similarly players from A either beat or get beaten by π', hence $|A| = \text{in}\,_A(\pi') + \text{out}\,_A(\pi')$. Plugging it together we infer

$$1 + \text{out}\,_A(\pi') + \text{out}\,_B(\pi') = \text{out}\,(\pi') \leq |A| = \text{in}\,_A(\pi') + \text{out}\,_A(\pi')$$

from which it follows

$$1 + \text{out}\,_B(\pi') \leq \text{out}\,_A(\pi').$$

This settles (2).

In what follows we shall give a recursive argument, that the vertices of $\rho \setminus \{\pi\}$ can be partitioned so that for every member of the partition, i.e. a subset of players, we can construct a binomial tree of a bracket with a player from A being the root.

Naively one could try a greedy approach. Choose a player $\pi' \in A$ and a collection of players $S \subseteq B$, so that π' beats every player from S and the cardinality of $S \cup \{\pi'\}$ is a power of 2. Clearly π' is a winner in every bracket having players $S \cup \{\pi'\}$. We would proceed by greedily choosing another player π'' from $A \setminus \{\pi'\}$ and a collection $S'' \subseteq B \setminus S$.

This naive approach may run us into problems. What one needs to check is that π'' is in similar relation to $B \setminus S$ that π' was to B, in order to get the recursion to end. Formally, the solution lies in the next claim.

(3) Let $A' \subseteq A$ and $B' \subseteq B$ be nonempty sets such that for every player $\pi' \in B'$ we have $\text{out}_{B'}(\pi') < \text{in}_{A'}(\pi')$. Then we can pick a player $\pi^* \in A'$ and a subset of players $S \subseteq N^{\text{out}}_{B'}(\pi^*)$ so that

(i) $|S \cup \{\pi^*\}|$ is a power of two and

(ii) for all $\pi'' \in B' \setminus S$ we have

$$\text{out}_{B' \setminus S}(\pi'') < \text{in}_{A' \setminus \{\pi^*\}}(\pi'') .$$

Pick an arbitrary player π^* of A' that has an outneighbour in B'. Such a player exists, as by assumption every player from B' has a positive indegree.

Let us now choose integers k, r with k maximal possible so that $|N| = 2^k - 1 + r$. We will show that in this case

$$|N| \geq 2r.$$

If $|N| < r$ and k maximal possible, then $|N| = 2^k - 1 + r < 2r$ implies that $2^k \leq r$. Hence we can rewrite

$$|N| = 2^k - 1 + 2^k + r_0,$$

where $r_0 = r - 2^k \geq 0$. This implies

$$|N| = 2^{k+1} - 1 + r_0$$

which is a contradiction to the maximality of k. We infer that

$$|N| \geq 2r \quad \text{and} \quad 0 \leq r \leq 2^k - 1.$$

We need at least $2r$ vertices to get r edges in a matching. Choose an arbitrary matching M of size r in N, and let R be the r heads of arcs from M. Finally let $S = N \setminus R$. Obviously enough π^* beats every player of S.

We claim that π^* and S satisfy both (i) and (ii).

$|S| = |N| - |R| = (2^k - 1 + r) - r$ and $|S \cup \{\pi^*\}| = 2^k$ which settles (i).

For (ii) let us choose an arbitrary player $\pi'' \in B' \setminus S$. If $\pi'' \notin N$, then its indegree from A' does not change when removing π^*. We have

$$\mathrm{in}_{A' \setminus \{\pi^*\}}(\pi'') = \mathrm{in}_{A'}(\pi'') > \mathrm{out}_{B'}(\pi'') \geq \mathrm{out}_{B' \setminus S}(\pi'')\,.$$

On the other hand if $\pi'' \in N$ then $\pi'' \in R$, as $\pi'' \in B' \setminus S = B' \setminus (N \setminus R)$. Now π'' is a head of an edge in matching M, so its outdegree in $B' \setminus S$ drops by at least 1 compared to its outdegree in B'. Hence

$$\mathrm{out}_{B' \setminus S}(\pi'') \leq \mathrm{out}_{B'}(\pi'') - 1 < \mathrm{in}_{A'}(\pi'') - 1 = \mathrm{in}_{A' \setminus \{\pi^*\}}(\pi'')$$

where the last equality follows at π^* beats π''. This proves (3).

Now the reasoning goes as follows. Assume that π satisfies (STR). By (2) we see that the starting conditions of (3) are satisfied initially with $A' = A$ and $B' = B$. We shall use (3) inductively to output a sequence of pairs

$$(\pi_1, S_1) \ldots (\pi_\ell, S_\ell)$$

so that $A = \{\pi_1, \ldots, \pi_\ell\}$ and $B = S_1 \cup \ldots \cup S_\ell$. Note that S_i may be empty and also that if B' is nonempty then also A' is nonempty in the starting condition of (3). This sequence is used to produce a collection of binomial trees rooted at $\pi_1, \ldots \pi_\ell$, and (1) proves the existence of a binomial tree containing all vertices of ρ in which π is the winner.

Sections 4 and 5 are conceptually different. We do not construct *binomial trees*, but we construct *binary trees* inductively. First we construct a matching in the first round and then proceed inductively by preserving a heuristic property at the next round.

2.4 Kings who beat half the players

In order to prove the (KNG) case of Theorem 2.1, let us fix a player π, and assume that he matches condition (KNG).

Let us first observe the very small cases. If $|\rho| = 1$ then π is the only player in ρ and no matches are needed. If $|\rho| = 2 = 2^1$, then the only remaining player π' clearly loses in a match with π, because according to (KNG) case we have

$$|\, N^{\text{out}}\,(\pi)| \geq |\rho|/2 = 1$$

and the only possible outneighbour of π is π', which makes π a winner.

Let us turn our attention to bigger collections of players. We will henceforth assume that $|\rho| = 2^k \geq 4$, or equivalently $k \geq 2$. We can also assume that π is not the Condorcet winner.

We will construct a bracket for which π emerges as the winner of the binary competition tree. Let $A = N^{\text{out}}\,(\pi)$ be the set of players who get beaten by π, and let $B = N^{\text{in}}\,(\pi)$ be the set of players which beat π. Now (A,B) is a partition of $\rho \setminus \{\pi\}$, where both A and B are nonempty sets of players. The set B is nonempty as by assumption π is not the Condorcet winner, and given an arbitrary $\pi' \in B$ we have (by (KNG)) also π'' who beats π' and loses to π implying that also A is nonempty.

Let us construct a *maximal matching* M_{AB} from A to B so that the matching includes at least one pair as A and B are nonempty and π is a king (by (KNG)). In particular, there is a player π_A in $N^{\text{out}}\,(\pi)$ who beats all players in $N^{\text{in}}\,(\pi)$ so that all players in $N^{\text{out}}\,(\pi) \setminus \{\pi_A\}$ lose to every player in $N^{\text{in}}\,(\pi)$. We can pick π' in A to play against π in the first round, because

$$|A| = \text{out}\,(\pi) \geq |\rho|/2 \quad \text{and} \quad |B| = |\rho| - \text{out}\,(\pi) - |\{\pi\}| < |\rho|/2.$$

Finally, given that we have remaining vertices, we can construct two *perfect matchings* $M_{A'}$ and $M_{B'}$ on $A' = A \setminus (V(M_{AB}) \cup \{\pi'\})$ and $B' = B \setminus V(M_{AB})$ respectively, because the number of remaining vertices in A' and B' altogether is either 0 or 2^k, $k \geq 1$. But, if $|B'|$ is odd, then the number of players in A' is also odd, because $|\rho| - (2|M_{AB}| + |\{\pi, \pi'\}|)$ is even, and in this case we match a player from A' with a player from B', and then make $M_{A'}$ and $M_{B'}$.

Further assuming that ρ' is the collection of players who survived to the second round, we need to prove that π satisfies (KNG) for ρ'. Namely, we will show that

(i) $\text{out}\,(\pi) \geq |\rho'|/2$ and

(ii) π is a king in ρ'.

Let us set $n = |\rho|$ and let m denote the number of vertices of B in M_{AB}. The players who can beat π in ρ' are sources of edges in M_B. If $|B| - m$ is even there are $\frac{|B|-m}{2}$ of them. On the other hand, these players are not only the sources of edges in M_B but also the player from B' matched with a player from A' if $|B| - m$ is odd. There are $\left\lfloor \frac{|B|-m}{2} + 1 \right\rfloor$ of such players in this case. We infer that in both cases there are $\left\lceil \frac{|B|-m}{2} \right\rceil$ players from B who survive to the second round. Recall that $m \geq 1$. This implies that

$$\left\lceil \frac{|B| - m}{2} \right\rceil \leq \left\lceil \frac{|B| - 1}{2} \right\rceil \leq \frac{(\frac{n}{2} - 1) - 1}{2} = n/4 - 1 < n/4.$$

Now let B_2 and A_2 denote the number of in- and outneighbours of π in ρ' respectively. We have $|B_2| < n/4$ and consequently $|B_2| \leq n/4 - 1$ which implies that

$$|A_2| \geq n/4.$$

Recall that $n/2$ is exactly the number of players in ρ' and hence

$$\text{out}\,(\pi) \geq |\rho'|/2$$

which settles (i).

Next if π_{B_2} is a player from B_2 who survived the first round, then $\pi_{B_2} \in \rho' \cap B_2$. Hence, there exists a player $\pi_{A_2} \in A_2$ who beats π_{B_2}, because π is a king (by (KNG)). Assuming that π_{A_2} does not survive the first round and $\pi_{A_2} \notin \rho'$, we have a contradiction with the maximality of M_{AB}, because $M_{AB} \cup \{\pi_{A_2}, \pi_{B_2}\}$ is a bigger matching. Hence, $\pi_{A'}$ survives round one and this settles (ii).

Now inductively we infer that (i) and (ii) hold for $\log_2(n) + 1$ rounds so that there exists a *binary competition tree* which includes all vertices from ρ with π emerging as the *winner*.

Now let us exhibit an example showing that the bound in (KNG) is tight. If π is a king and only beats $|\rho|/2 - 1$ players then it may not be possible to make him a winner.

Let us partition $\rho \setminus \{\pi\}$ into sets A and B, so that $|A| = |\rho|/2 - 1$ and $|B| = |\rho|/2$. Assume that π beats every player from A, and loses to every player from B. Next, choose a player $\pi' \in A$ and assume that π' beats all players from B. These results make π a *king*, as for every player $\pi^* \in B$, who wins over π, we have π' who loses to π and beats π^*. Yet, π does not satisfy (KNG) as he only beats $|\rho|/2 - 1$ players.

Let us also assume that every player from B beats every player from $A \setminus \{\pi'\}$, and choose the remaining outcomes arbitrarily. We claim that π cannot be a winner.

Assume to the contrary that π is a winner, and let T be the binary tree of a bracket with π as the root. Let T' be the subtree of T rooted at π'. As T' is a proper subtree of T, it contains strictly less than $|\rho|/2$ nodes from B. The remaining players from B, as they beat all players from $(A \cup \{\pi\}) \setminus \{\pi'\}$, can propagate to the top, which contradicts to π being a winner.

In the next section we consider a special case of a *king* and repeat the steps to construct a *bracket*, but without setting an *outdegree* limitation on π.

2.5 Super-kings

In order to prove the (SKG) case of Theorem 2.1 we fix a player π from ρ who satisfies (SKG) and is a *super-king*.

Let us first consider small and easy cases. If $|\rho| = 1$ it is obvious that π is a winner and no matches are needed. On the other hand, if $|\rho| = 2$ then also π emerges as a winner. Recall that every *super-king* is a *king*. Assuming that $\pi' \in \rho$ and π' beats π, we have $\pi' \in N^{\text{in}}(\pi)$. But, there should be a player π'' who beats π' and loses to π as π is king. But, ρ includes only 2 players, hence $\pi' \in N^{\text{out}}(\pi)$ which makes π a winner in this competition.

Having treated the basic cases we turn out attention to a collection of players ρ, where $|\rho| = 2^k \geq 4$, or equivalently $k \geq 2$. We can also assume that π is not the Condorcet winner.

We will construct the first round of our competition so that π proceeds to the second round. Also, we will prove that (SKG) condition holds for π in the second round as well.

Let us partition $\rho \setminus \pi$ into two subsets A and B, out- and inneighbours of π, respectively. Now (A,B) is a partition of $\rho \setminus \{\pi\}$, where both A and B are nonempty sets of players: by the assumption that π is not the Condorcet winner B is nonempty. Hence, also A is nonempty as π is a king and given an arbitrary $\pi' \in B$ we have $\pi'' \in A$ who beats π'.

We fix a player π' in A to play against π in the first round so that π survives to the second round. This should not be a problem as every player π_B from B has at least $\log_2(4) = 2$ players from A who beat π_B (recall that $|\rho| \geq 4$). Next we choose a *maximal matching* M_{AB} from A to B, and this matching is nonempty (here we are using the fact that A and B are nonempty). Further we can find a *perfect matching* M which extends $\{\pi, \pi'\} \cup M_{AB}$ by matching the remaining players in A and B, because $|\rho| - |\{\pi, \pi'\}| - 2|M_{AB}|$ is even.

Let ρ' be the collection of players who survive to round two — these are exactly the sources of edges in M. Also, let A_2 and B_2 be out- and inneighbours of π in ρ' respectively. Assume that $\pi_B \in B$ survived to the second round and hence π_B is also from $B \setminus V(M_{AB})$. As M_{AB} is maximal there is no player in $A \setminus (V(M_{AB}) \cup \{\pi'\})$ who beats π_B. All players who beat π_B are in $V(M_{AB}) \cup \{\pi'\}$, so that all of them but π' survive to the second round. Hence, π_B has at least $\log_2(|\rho|) - 1$ players in A_2 who beat him. As $|\rho|/2$ players from ρ survive to round two and $\log_2(|\rho|) - 1 = \log_2(|\rho|/2)$, we infer that for every player $\pi_{B_2} \in B_2$ there are at least $\log_2(|\rho'|)$ players from A_2 who beat π_{B_2}. This shows that π satisfies (SKG) in round two as well.

We proceed by induction for $n = 2^k$ players in ρ, where $k > 2$, and hence for $\log_2(n)$ rounds so that there exists a *binary competition tree* where π is the *winner*. This finishes (SKG) proof of Theorem 2.1.

2.6 A winner not satisfying Williams cases

In the previous sections we have proved Theorem 2.1 separately for each of the cases (STR), (KNG) and (SKG). Given a player $\pi \in \rho$ who satisfies one or several of the sufficient conditions, there exists a *bracket* so that π is the *winner* of the *competition tree*. But, what if π does not satisfy (STR), (KNG) and (SKG). Can π still win a competition? We will present an example of a binary competition tree where π does not satisfy either of Williams' conditions, but π still emerges as the winner of the tree.

Consider a collection of players $\rho_2 = \{0, 1, 2, 3, 4, 5, 6, 7\}$. The partial collection of results, matrix R_{ρ_2}, is depicted in Table 2.1.

	0	1	2	3	4	5	6	7
0	X	1	1	0	1	0	0	0
1	0	X						0
2	0		X	1				0
3	1		0	X				0
4	0				X	1	1	0
5	1				0	X		0
6	1					0	X	1
7	1	1	1	1	1	1	0	X

Table 2.1: Matrix R_{ρ_2} showing results of matches between players from ρ_2.

Let us consider matrix R_{ρ_2}. The outcomes of matches of players 0 and 7 are completely determined. The matrix contains a number of blank spots where the direction of outcome can be chosen arbitrarily. Player 0 does not satisfy (STR), because

$$3 = \text{out}\,(0) < \text{out}\,(7) = 6.$$

Also, player 0 does not satisfy (KNG), because he/she is not a *king* who beats half the players in ρ_2, as

$$3 = \text{out}\,(0) < |\rho_2|/2 = 4.$$

Finally, for player 7 who beats 0 there are no $\log_2(|\rho_2|) = \log_2(8) = 3$ players who beat 7, and hence player 0 is not a *super-king* and does not satisfy (SKG). We infer that player 0 is not a *W-winner*. But, we can construct a binary competition tree with player 0 emerging as the winner, see Figure 2.1.

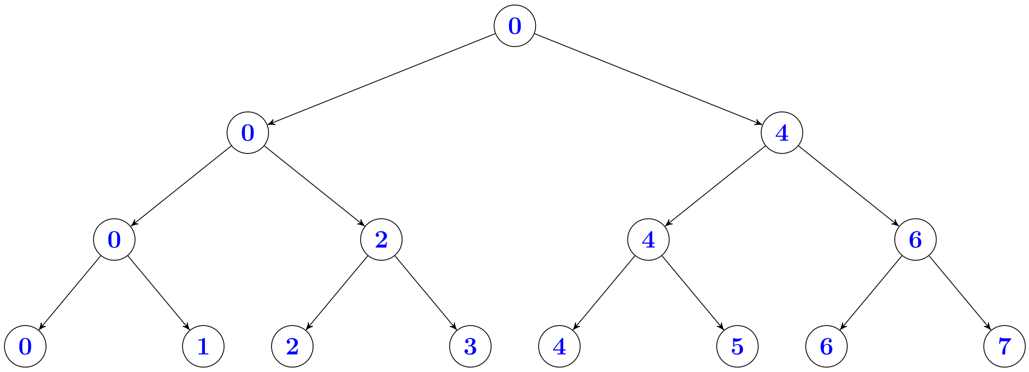

Figure 2.1: A *binary competition tree* T_{β_0} with not a *W-winner*.

Let us consider T_{β_0}. We fix player 1 to play against player 0, and we knock-out players 3 and 5, the inneighbours of player 0, by players 2 and 4 respectively, in the first round. Note that player 7 loses to player 6 in the first round. Next, player 0 meets the outneighbour player 2 in the second round, and also player 4 beats player 6, the inneighbour of player 0. Hence, player 0 meets player 4 in the final of our competition. This makes player 0 the winner in T_{β_0}. Note that we can set the absent results of players $1 \ldots 6$ arbitrarily so that player 0 still emerges as a winner in bracket β_0.

It is not difficult to show that this example is the smallest possible. In other words, for every collection of players ρ, $|\rho| = 4$, a winner always satisfies (STR) or (KNG) or (SKG), see Figure 2.2.

The outdegree of player 0 is not less than the outdegree of player 3, despite the outcome of a match between players 1 and 3, and hence player 0 satisfies case (STR). Also, player 0 is a king who beats half the players and consequently satisfies case (KNG). But, if $|\rho| > 4$ then this may be not

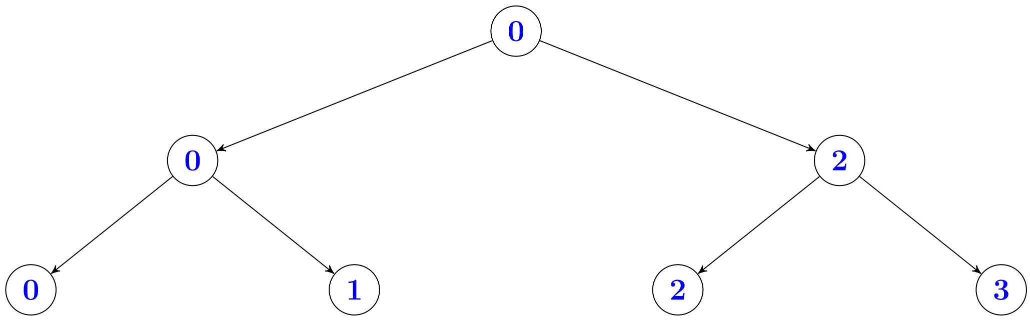

Figure 2.2: T_{β_0} with the smallest $|\rho| = 4$ for a *W-winner*.

possible for π to satisfy at least one of the conditions, and we have shown this in Figure 2.1.

In the next chapter we make tests to find possible winners who are not *W-winners* in realistic situations and see how good Williams cases (STR), (KNG) and (SKG) are.

Chapter 3

Tests

In this chapter we discuss our data and finish by showing results of three tests that we performed .

3.1 Data

The Association of tennis professionals, or ATP, is an international governing body gathering players, promoters, managers and competition organizers. It also publishes regulations and keeps tennis present in the media world [1].

Organizers of ATP competitions have to follow the published calendar [2] and provide sufficient money prize lots. The biggest competitions include the four Grand Slams: Australian Open, Rolland Garros, Wimbledon and US Open. An example of a lower ATP World Tour competition is St. Petersburg Open.

The best tennis players of the world essentially only take part in ATP competitions, and ATP also keeps track of the history of their matches and publishes the ranking of players. Our first collection of data was obtained from [3] in winter 2014/2015 where the data set contained the match history of 103 players. Later, in June 2015, the ATP database was updated and over 1000 players were present in the ranking [4]. Our second data set contained the match history of 128 players. Finally, in September 2015 we acquired the third data set which contained the match history for 148 players, and we could provide more realistic tests with *brackets* of 128 players.

The ATP ranking is dynamic and is updated after every finished competition. The players are ranked according to their success at various levels of competitions in the previous 365 or 366 days. There are significant differences between the first two data sets. For instance, Rafael Nadal moved down by 7 positions and became 10th in June 2015. In the same period, Mikhail Youzhny moved down by 29 positions, from 47th to 76th, yet Mikhail Kukushkin moved by a single place.

We used a service called ATP.HEAD2HEAD [1], which includes a brief history of matches between two given players. Let us consider Novak Djokovic and Marin Cilic history of matches up to September 13 of 2015. It is highly probable that Djokovic wins over Cilic as Djokovic has won 13 times over Cilic and has never lost. So in our scores table R we indicate that Djokovic wins over Cilic. On the other hand, in a match between Djokovic and Federer it is not evident who is the winner, because Djokovic has won 20 times and lost 21 times to Federer[1]. But, we still use the history of matches and indicate in R that Federer is the winner. This might not be the case in real, and Djokovic may emerge as the winner. But, it may also not be the case that two players met each other in ATP competitions.

We have a problem in a pair Steve Darcis and Alexander Zverev. They have so far not played a match in ATP. In this case we have opted to look at their career prize money — the total amount up to June 2015. We proclaim Darcis as the winner, because Zverev earned 492,289$, and Darcis 2,087,918$. We realize that such a variant of constructing R may not be flawless. If two players are at the start of their careers then the more experienced player is more likely to win. Recall that ATP pays prize money in a competition according to the round, and the semi-final player earns more than the quarter-final player who did not reach the semi-final. Note that it never was a case that two players earned exactly the same amount of prize money.

[1]Federer has just lost in US Open and the score is 21 - 21, but not in our data set.

Let us consider the scores table R we obtained by September 2015. The history of match results of the first 11 players in our ranking shows that each two players met in ATP competitions. But, starting from player ranked 12 we have some gaps in our data. There are two players who never met in ATP competitions. We may say these players under 12 are the top ones who often survive to later rounds in competitions. This also implies that they play more matches on average than other players.

As the top players on average play more matches it is not surprising that the densest part of our match results is the range of players between 1 and 40. On the other hand, players ranked below 69 rarely met in ATP competitions, and most of the results are defined according to money prizes. We realize that there are players who recently started their careers, and players who will finish their careers soon. In most cases the younger players have accumulated less money prizes than the older players. But, it is not evident that a younger player should lose to the older player, as the younger player may be talented and is potentially physically stronger than the older player.

In the next sections we finally test our data and hope to acquire useful results.

3.2 How low can *W-winner's* rank be?

We would like to find as many W-winners as possible. But, we are more interested if a weak player can be a W-winner. In other words, how weak a player $\pi \in \rho$ may be so that there exists a *bracket* for which π is a *W-winner*?

We have used the data set three from the beginning of September 2015. In our test we have removed top twenty players so that there are no *Condorcet winners* in ρ. We have created 10.000 random combinations of 64 players from data set three that included 148 players. Separately, for each of the cases (STR), (KNG), and (SKG) we have computed W-winners for each combination of players. For both the ATP ranking and the outdegree ranking we have done the following. Given a single rank r we have computed how many times a player with relative rank r satisfies at least one of (STR), (KNG), or (SKG) conditions. The histogram with results is depicted in Figure 3.1 [7].

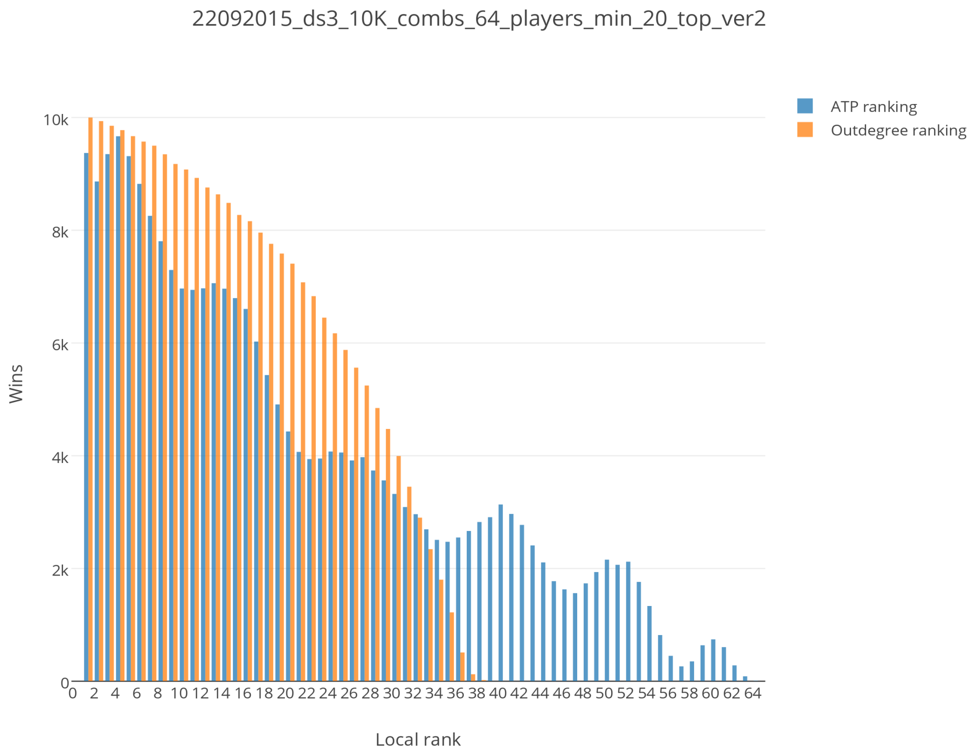

Figure 3.1: Frequences of *W-winners* among 10.000 collections of players.

On average, players with local ranks between 1 and 36, according to the outdegree ranking, can be winners. To give a qualitative indication, there is an approximately 5% probability that the player whose local rank is 36 is a W-winner. The probability drops to below 1% at the player whose rank is 38.

Consider the following scenario. We manage a player π whose rank is 100, and we also have connections with competitions' organizers so that we can rig the brackets in $\pi's$ favor. Which is the richest possible competition we should send our player π to in order to make him a winner. We should select a competition so that no very strong player appears there and our player π is slightly below the average rank in the competition.

Figure 3.1 indicates that in the relative ATP ranking all players between 1 and 64 satisfied one of (STR), (KNG) and (SKG) conditions — not in all cases but with a sufficiently high probability.

Let us try to give an explanation. The ATP ranking only takes the results of a single previous year into account whereas our scores include all historic results of a player. For instance, Lleyton Hewitt, currently is ranked number 101[2] on the ATP list, was once the world's top player[3]. Hence, his historic results make him a very strong player according to our scores. This implies that Hewitt is a W-winner whenever there are no currently very strong players in the competition. His local ATP rank can, of course, be arbitrary if the remaining players are chosen appropriately. Two further such players are Mikhail Youzhny and Albert Montanes. These players are the reason for local drops and rises in the histogram.

3.3 How good is Williams?

In this section we tackle the following question. Given realistic data, can we find a winner of a competition who is not a W-winner?

In section 2.6 we have constructed a collection of players, results, and a *bracket* in which a winner did not satisfy either of (STR), (KNG), or (SKG) conditions. But, do such winners appear in a practical situation?

For the experiment we have chosen a collection of 64 players: we have used data set three, from the beginning of September 2015, which contains 148 players in total. In order to exclude *Condorcet winners* we have decided to skip the first 20 players in the ATP ranking.

Next, we have made 10.000 permutations of ρ in order to construct binary competition trees for real winners. We have translated each real winner into

[2] 11th of September 2015.
[3] In years 2001 and 2003.

ATP and outdegree ranks and computed frequences of wins of the player, separately, for the ATP and the outdegree ranks. The result is depicted in Figure 3.2.

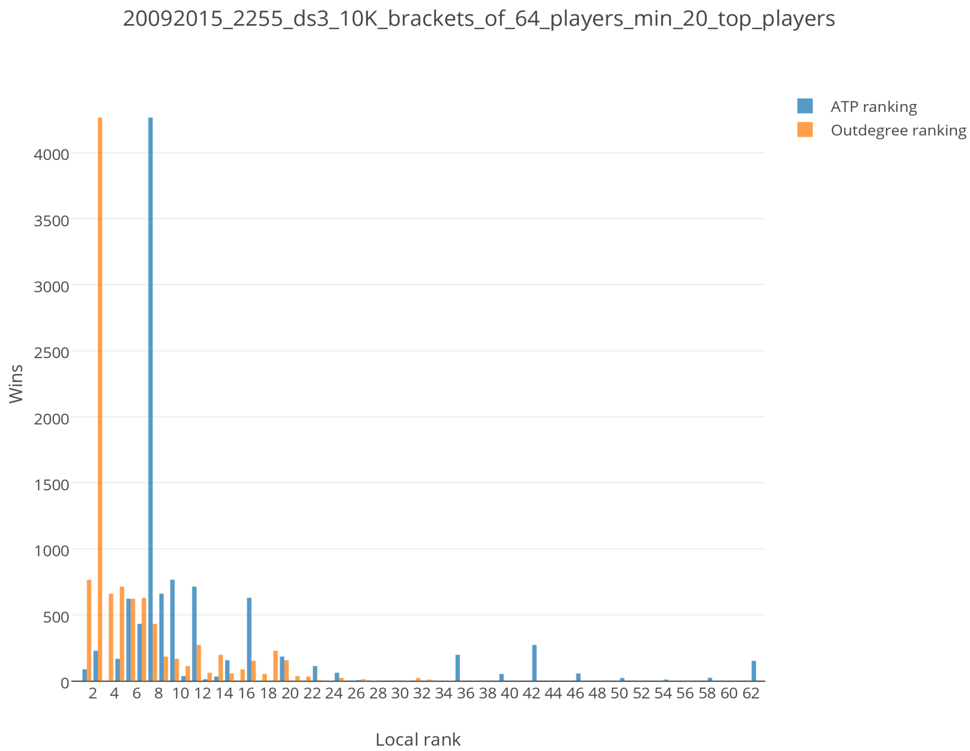

Figure 3.2: Real winners of 10.000 permutations of ρ.

In Figure 3.2 we have 33 real winners. Table 3.1 describes frequences of wins of the last 10 winners according to the outdegree ranking.

Local outdegree rank	23	24	26	27	30	31	32	33	34	36
Wins #	2	24	14	3	4	23	11	1	3	1

Table 3.1: Wins of the last 10 real winners by the outdegree ranking.

We indicate that there is 0.24% probability that a player with a local outdegree rank 24 will be a winner in our working collection of players. Also, the probability drops to 0.01% for the player ranked 36.

For all winners we also checked that they satisfy at least one of (STR), (KNG), or (SKG) conditions. We can state: with a high probability there are no winners who are not W-winners in realistic cases.

3.4 The random case

As the last test we focus on random data. For a collection of players ρ we produce R by flipping a fair coin, uniformly, independently at random. This states that the probability of $r_{i,j}$ being 1 is 1/2.

We have constructed random scores 100 times, and every time we compute the *W-winners*, players who satisfy (STR), (KNG) or (SKG).

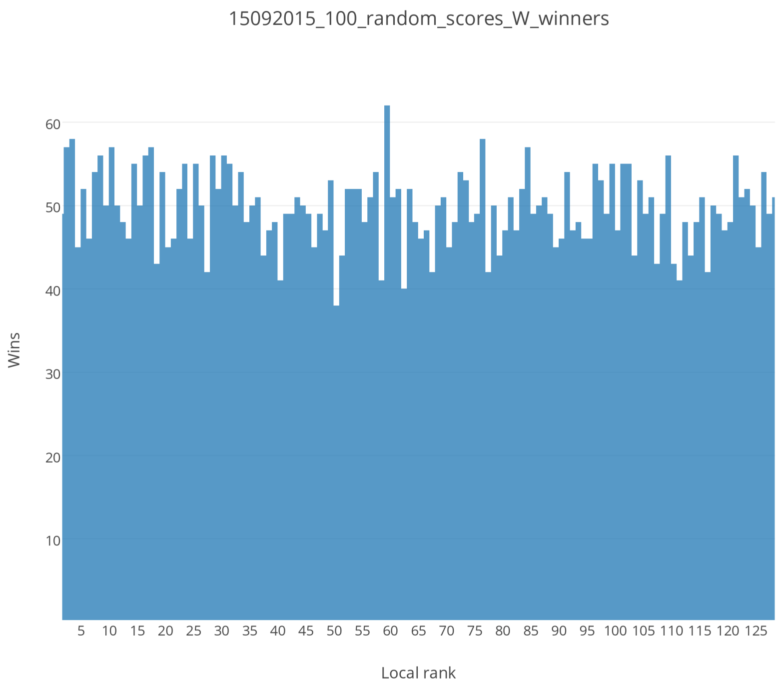

Figure 3.3: W-winners for 100 random scores and one ρ.

Figure 3.3 shows that in the random world every player is a W-winner. It is not difficult to argue why. We claim that with high probability an arbitrary player π satisfies (SKG). On average, π wins over approximately $|\rho|/2$ players, and also loses too approximately $|\rho|/2$ players, with fairly small (of the order $O(\sqrt{n})$) standard deviation.

Next, given a player π' we have on average $|\rho|/4$ players who lose to π and win over π'. This implies that π is very likely a *super-king*, because for large $n = |\rho|$ we have $\frac{1}{4} \cdot n > \log_2(n)$, or in other words, for every player π' winning over π there are more than $\log_2(n)$ players who lose to π and beat π'.

As a conclusion, we can state that ATP rankings definitely are not random.

Bibliography

[1] ATP, "Atp official page," [Online; accessed 19-September-2015]. [Online]. Available: http://atpworldtour.com

[2] ——, "Atp calendar," [Online; accessed 19-September-2015]. [Online]. Available: http://www.atpworldtour.com/en/tournaments

[3] ——, "Atp singles ranking, men," [Online; accessed 19-September-2015]. [Online]. Available: http://legacy.tennis.com/players/players_atp.aspx/

[4] ——, "Atp singles ranking, men," [Online; accessed 19-September-2015]. [Online]. Available: http://www.atpworldtour.com/en/rankings/singles

[5] R. Chen and F. K. Hwang, "Stronger players win more balanced knockout tournaments." *Graphs and Combinatorics*, vol. 4, no. 1, pp. 95–99, 1988. [Online]. Available: http://dblp.uni-trier.de/db/journals/gc/gc4.html#ChenH88

[6] N. Hazon, P. E. Dunne, S. Kraus, and M. Wooldridge, "How to rig elections and competitions," in *In Proc. of the 2nd Int. Workshop on Computational Social Choice (COMSOC)*, 2008.

[7] A. Johnson, C. Parmer, J. Parmer, and M. Sundquist, "Plotly," [Online; accessed 19-September-2015]. [Online]. Available: http://plot.ly

[8] J. Lang, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh, "Winner determination in sequential majority voting." in *IJCAI*,

M. M. Veloso, Ed., 2007, pp. 1372–1377. [Online]. Available: http://dblp.uni-trier.de/db/conf/ijcai/ijcai2007.html#LangPRVW07

[9] P. K. Newton and K. Aslam, "Monte carlo tennis," *SIAM Rev.*, vol. 48, no. 4, pp. 722–742, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1137/050640278

[10] P. K. Newton and J. B. Keller, "Probability of winning at tennis i. theory and data." *Studies in Applied Mathematics*, vol. 114, no. 3, p. 241–269, Apr. 2005.

[11] I. Stanton and V. V. Williams, "Rigging tournament brackets for weaker players," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, ser. IJCAI'11. AAAI Press, 2011, pp. 357–364. [Online]. Available: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-069

[12] A. Syvak, "Bsc thesis home page," [Online; accessed 20-September-2015]. [Online]. Available: http://bsc.syvak.besaba.com

[13] W. T. Tutte and C. S. J. A. Nash-Williams, *Graph theory*, ser. Encyclopedia of mathematics and its applications. Cambridge, UK, New York: Cambridge University Press, 2001, vol. 21. [Online]. Available: http://opac.inria.fr/record=b1101974

[14] T. Vu, A. Altman, N. Hazon, S. Kraus, Y. Shoham, and M. Wooldridge, "On the complexity of schedule control problems for knock-out tournaments," in *Submitted to JAIR*, 2010.

[15] T. Vu, A. Altman, and Y. Shoham, "On the complexity of schedule control problems for knockout tournaments," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS '09. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009,

pp. 225–232. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558013.1558044

[16] J. Vuillemin, "A data structure for manipulating priority queues," *Commun. ACM*, vol. 21, no. 4, pp. 309–315, Apr. 1978. [Online]. Available: http://doi.acm.org/10.1145/359460.359478

[17] V. V. Williams, "Fixing a tournament." in *AAAI*, M. Fox and D. Poole, Eds. AAAI Press, 2010. [Online]. Available: http://dblp.uni-trier.de/db/conf/aaai/aaai2010.html#Williams10