5.3 Arithmetization: One-Player Games with Randomized Transition

In section 3 Games, to win, the player (call him Merlin), must beat another perfectly powerful wizard (say, Lady of the Lake). To make the challenge realistic we remove the Lady, giving Merlin to evaluate, rather than choose, moves. Its correctness is challenged by a randomized transition, performed by a simple player, Arthur. He chooses moves for which Merlin's assessment is almost surely wrong unless it is correct for all moves. The wrong assessment will not match the assessment after the next move, unless it is wrong, too, etc.

The trick achieving this, called ***arithmetization***, was proposed in Noam Nisan's article widely distributed over email in the Fall of 1989 and quickly used in a flood of follow-ups for proving relations between various complexity classes. We follow [Shamir 90, Fortnow, Lund 93]. The trick is based on the feature that degree d polynomials coincide on the whole field if they do on more than d points. To use it we express the boolean functions as low degree polynomials, and apply them to $\mathbb{Z}_p$-tokens (let us call them ***bytes***) instead of bits.

This reduces generic games to games in which any Merlin's strategy in any losing configuration has exponentially small chance to win. The reduction holds for games with any (exponential, polynomial, etc.) limit on the remaining moves counter c. This c will be included implicitly in games configurations below.

Let g be the (ATM-complete) game of 1d-Chess (3.3), $r(m,x)$ with $x{=}x_1 \dots x_s$, $m, x_i{\in}\{0,1\}$ be its transition rule. Configurations include x and a remaining moves counter $c \le 2^s$. They are terminal if $c{=}0$, winning to the player x_1. Intermediate configurations (m,x,y) have y claimed as a prefix of $r(m,x)$.

Let $t(m,x,y)$ be 1 if $y{=}r(m,x)$, else $t{=}0$. 1d-Chess is simple, so t can be expressed as a product of s multilinear $O(1)$-sized terms, any variable shared by at most two terms. Thus t is a polynomial, quadratic in each m, x_i, y_i. Let $V_c(x)$ be 1 if the active player has a strategy to win in the c moves left, i.e. $V_0(x){:=}\,x_1$, $V_{c+1}(x){:=}\ 1{-}V'_c(0,x,\{\})V'_c(1,x,\{\}) = 1{-}V_c(r(0,x))V_c(r(1,x))$, where $V'_c(m,x,y){:=}\,V_c(y)t(m,x,y)$ for $y = y_1 \dots y_s$ or $V'_c(m,x,y){:=}\,V'_c(m,x,y{\circ}0){+}V'_c(m,x,y{\circ}1)$ for shorter y. ($\circ$ stands for concatenation.)

G will allow Merlin to prove x is winning i.e., $V_c(x) = 1$. At the start Merlin chooses a $2s$-bit prime p. Configurations $X = (m,x,y,v)$ of G replace x_i, m, y_i bits with $\mathbb{Z}_p$-bytes and add $v{\in}\mathbb{Z}_p$ reflecting Merlin's claimed $v{=}V'_c(m,x,y)$. The polynomial V_c retains the above inductive definition, thus is quadratic in each x_i, m, as $t(m,x,y)$ is. Then y_i have degree ≤ 4 in $V_c(y)$ and ≤ 6 in $V_c(m,x,y)$.

At his steps Merlin chooses a univariate polynomial P of degree 6. v must be $1{-}P(1)P(0)$ for s-byte y, or $P(0){+}P(1)$ for shorter y. Arthur then chooses a random $r{\in}\mathbb{Z}_p$ and X becomes $(r, y, \{\}, P(r))$, or $(m, x, y{\circ}r, P(r))$, respectively. For X with a correct v Merlin's obvious winning strategy is to always provide the correct P. If v is wrong then either $P(1)$ or $P(0)$ must be wrong, too. So P will differ from V and they can agree only on few (bounded by degree) points. Thus $P(r)$ will be correct only on exponentially small fraction of random r. So, the wrong v will propagate throughout the game until it becomes obvious at $c{=}0$. This gives any Merlin strategy an exponentially small winning chance.

This reduction of Section 3 games yields a hierarchy of Arthur-Merlin games powers, i.e. the type of computations that have reductions to $V_c(x)$ of such games and back. The one-player games with randomized transition rule r running in space linear in the size of initial configuration are equivalent to exponential time deterministic computations. If instead the running time T of r combined for all steps is limited by a polynomial, then the games are equivalent to polynomial space deterministic computations.

An interesting twist comes in one move games with polylog T, too tiny to examine the initial configuration x and the Merlin's move m. But not only this obstacle is removed but the equivalence to NP is achieved with a little care. Namely, x is set in an error-correcting code, and r is given $O(\log \|x\|)$ coin-flips and random access to the digits of x, m. Then the membership proof m is reliably verified by the randomized r.
See [Holographic proof] for details and references.