Rewriting, Automata, and Games

Johannes Waldmann

Institut für Informatik

Universität Leipzig, Germany

http://theopc.informatik.uni-leipzig.de/~joe/trs/

Terms and Rewriting

Signature $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \ldots \cup \Sigma_n$
where Σ_k contains function symbols of arity k
Set of all Σ-terms: Term(Σ)

Rewrite system R defines relation $\to_R$ on Term(Σ)

Example 1: $\Sigma_0 = \{D\}, \Sigma_2 = \{\cdot\}$, $R = \{(D \cdot x) \cdot y \to x \cdot (x \cdot y)\}$.

Example 2: $\Sigma_0 = \{e\}, \Sigma_1 = \{a, b\}$, $R = \{a(b(x)) \to b(b(a(a(x))))$.

Normal forms $\mathsf{Nf}(R) := \{t \mid \neg \exists t' : t \to_R t'\}$

successors, one step: $R(L) := \{t' \mid L \ni t \to_R t'\}$, many: $R^*(L)$

Tree Languages and Automata

Finite tree automaton A on signature Σ, with finite state set Q, accepting states Q', transitions of the form $f(q_1, \ldots, q_k) \to q$.

$L(A) := \{t \mid t \to_A^* q \in Q'\}$.

Such languages L are called regular, recognizable, or rational. We write $L \in$ RAT.

Most of results (and proofs) from rational word languages carry over (closure under Boolean operations, powerset construction, pumping property, decidability of emptiness and finiteness, ...)

Rewriting and Rationality: Normal Forms

For a left linear rewrite system R, we have $\mathsf{Nf}(R) \in$ RAT.

Example : $R = \{(D \cdot x) \cdot y \to x \cdot (x \cdot y)\}$.

(Exercise: show that R is terminating)

$\mathsf{Nf}(R) = \{D, D \cdot D, D \cdot (D \cdot D), \ldots\}$

accepted by automaton: $D \to q_D,\ D \to q_N,\ q_D \cdot q_N \to q_N$.

generated by grammar: $N \to D \mid D \cdot N$.

Rewriting and Rationality: Successor Sets

If $L \in$ RAT, what about $R^*(L)$?

Motivation: check properties of R-computations, like

- absence of errors: $R^*(L) \cap E = \emptyset$
- normalization: $L \subseteq R^{-*}(\mathsf{Nf}(R))$
- results obey specification: $R^*(L) \cap \mathsf{Nf}(R) \subseteq S$

If R is *ground* (rules do not contain variables), ...

If R is *right linear inverse monadic*
(rules of the form $t \to x \mid f(x_1, \ldots, x_k)$), ...

... then $L \in$ RAT $\Rightarrow R^*(L)$.

Rewrite Games

Let two persons do the rewriting! $t_0 \to_A t_1 \to_B t_2 \to_A t_3 \to_B \ldots$

- it is more fun
- it models reactive systems

game theoretic problems

- might be solved by the rewriting approach
- invite to develop new rewriting techniques

What is a Game?

finite deterministic zero-sum two-player games

no chance (dice), no hidden information (others' cards), no ties

$\Rightarrow$ exactly one of the players *has* a winning strategy

... but how to find him, and his strategy, efficiently?

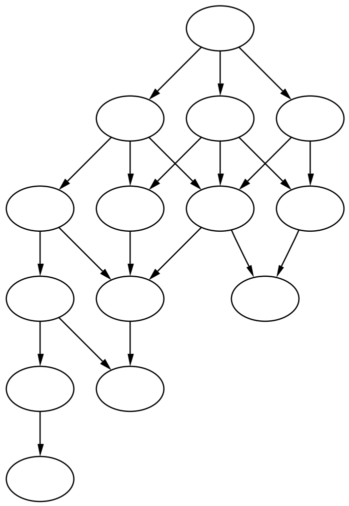
Combinatorial Game Theory (Bouton 1902, Lasker, Sprague $\approx$ 1930, Grundy 1939, Berlekamp, Conway, Guy $\approx$ 1980)

Normal Impartial Games

normal: the last player wins
(invalid move loses)

impartial: in each game position,
the set of available moves
does *not* depend on who's turn it is

view game as directed acyclic graph,
with one source (start position)



P/N-Labeling

For game position (digraph vertex) x:

$N(x) \iff$ next player has a winning strategy

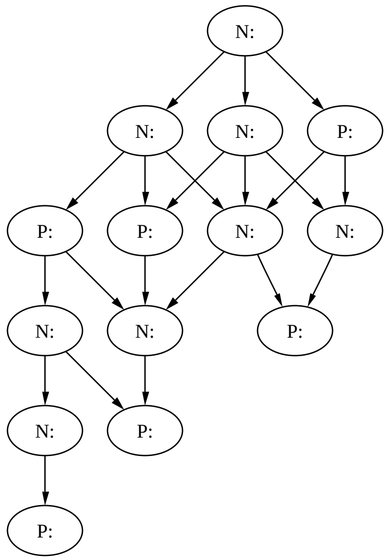
$P(x) \iff$ previous player has a winning strategy

$N(x) \iff \exists y : (x \to y) \land P(y)$

$P(x) \iff \forall y : (x \to y) \Rightarrow N(y)$.

$\Rightarrow$ two-colouring of the game digraph.

how complex can $\{x : P(x)\}$ get?



Take-and-Break Games

game position: multiset $H = \{H_1, \ldots\}$
of positive integers (heaps of beans)

rule set: list of sets $M = [M_0, M_1, \ldots, \emptyset, \ldots]$, example $[\emptyset, \{1, 2\}, \emptyset, \ldots]$

move: if $s \in M_k$, then: pick one heap, $\boxed{\text{take}}$ k beans,
and $\boxed{\text{break}}$ remaining heap into s (nonempty!) heaps.

formally: replace H by $H \setminus \{h\} \cup \{h_1, \ldots, h_s\}$
with $h_1, \ldots, h_s > 0$ and $h_1 + \ldots + h_s + k = h$

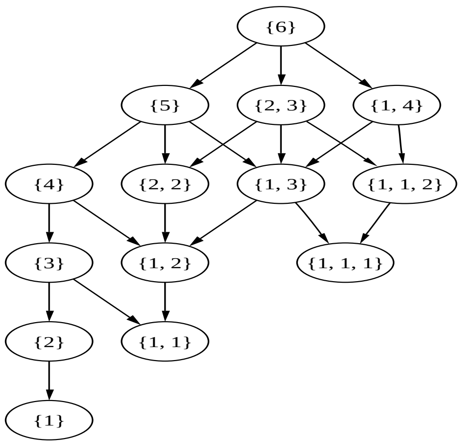
Example: $M = [\emptyset, \{1, \boxed{2}\}, \emptyset, \ldots]$, move : $\{1, \boxed{6}, 10\} \mapsto \{1, \boxed{1{,}4}, 10\}$.

Take-and-Break Game Example: Officers

ruleset $= [\emptyset, \{1, 2\}, \emptyset, \ldots]$

remove one bean, split remainder into one or two parts

promote a soldier to officer, assign at least one subordinate.



Octal Games

binary encoding of rule set $M = [M_0, M_1, \ldots, \emptyset, \ldots]$

replace M_k by $d_k = \sum_{s \in M_k} 2^s$

example: $[\emptyset, \{1, 2\}, \emptyset, \ldots]$ gives $[0, 2^1 + 2^2, 0, \ldots]$. Notation: 0.6

Def. A Take-and-Break game is *octal*
if $\forall k : 0 \le d_k < 8$ (i. e. $M_k \subseteq \{0, 1, 2\}$)

Heaps are split in at most two parts. $\Rightarrow$ string rewriting!

Rewrite Games (Definition)

Given a terminating rewrite system R, and a start term t_0, play the normal impartial game on the R-reduction graph of t_0.

In other words, two players A, B alternate in doing R-reductions $t_0 \to_A t_1 \to_B t_2 \to_A t_3 \to_B \ldots$.

The objective of the game is to get the last move — the player who cannot move (because t_k is in normal form), loses.

Play online example games at
`http://theopc.informatik.uni-leipzig.de/~joe/trs/`

Octal Games as String Rewrite Games

Alphabet: $\{0, 1\}$

encode game position $\{H_1, H_2, \ldots\}$ as string $0^+ \, 1^{H_1} \, 0^+ \, 1^{H_2} \ldots 0^+$

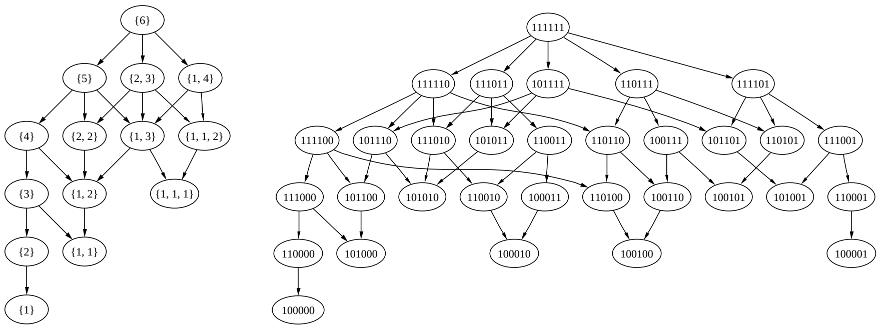
encode ruleset

- if $0 \in M_k$, then $01^k0 \to 0$
- if $1 \in M_k$, then $11^k0 \to 10, 01^k1 \to 01$
- if $2 \in M_k$, then $11^k1 \to 101$

combine some rules: officers $= 0.6 \cong \{11 \to 10\}$, James Bond $= 0.007 \cong \{111 \to 0\}$.

Octal Game → Rewrite Game, example

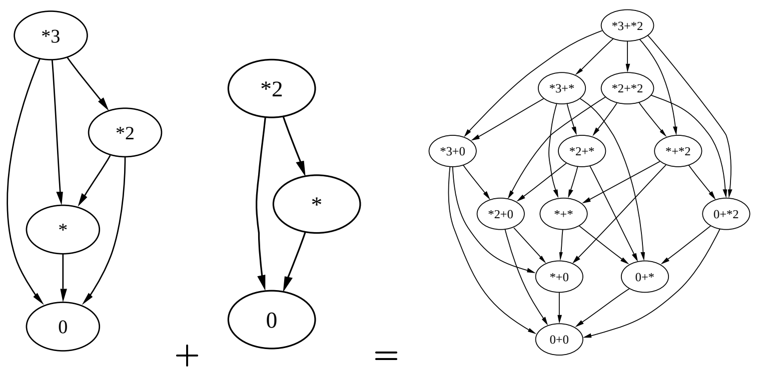
$[\emptyset, \{1, 2\}, \emptyset, \ldots] \approx 0.6 \approx \{11 \to 10\}$



Disjoint Sums of Games

$G + H$: (move in G and keep H), or (keep G and move in H).

$G + H := \{g + H : g \in G\} \cup \{G + h : h \in H\}$.



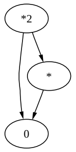
Game Equivalence (first try)

Def. $G \sim G' :\iff P(G) \leftrightarrow P(G')$.

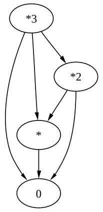
Observation. There are games G, G', H with $G \sim G'$, but $(G + H) \not\sim (G' + H)$.

$G =$

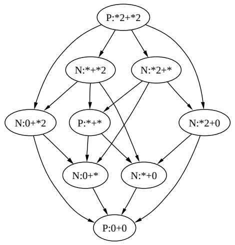
$H =$



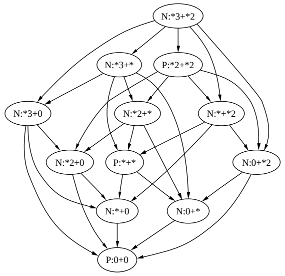
$G' =$



$G + H =$



$G' + H =$



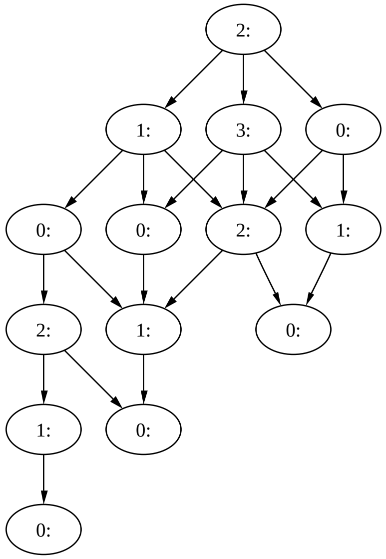
Grundy-Labeling

grundy(x)
:= mex{grundy(y) : $x \to y$}

mex M = minimal excluded value = min($\mathbb{N} \setminus M$).

"greedy" bottom-up game digraph labelling:

each node gets lowest colour available



Game Equivalence (stable version)

Def. $G \approx G' :\iff \forall H : P(G + H) \iff P(G' + H)$.

Theorem (Sprague, Grundy) $G \approx G' \iff$ grundy(G) = grundy(G').

Proposition: grundy($G + H$) = grundy(G) $\oplus$ grundy(H), where $\oplus$ is binary addition without carries.

(This solves standard Nim = 0.3333...)

Computation of Grundy Numbers

for Officers = 0.6 $\equiv [\emptyset, \{1, 2\}, \emptyset, \ldots]$.

use notation $G \approx *n \iff$ grundy(G) $= n$
and rules $\{*g_1, \ldots, *g_n\} \approx *g \iff g =$ mex$\{g_1, \ldots, g_n\}$
and $*g + *h = *(g \oplus h)$

$H_0 = \emptyset \approx *0, \quad H_1 = \emptyset \approx *0, \quad H_2 = \{H_1\} \approx \{*0\} \approx *1$

$H_3 = \{H_2, H_1 + H_1\} \approx \{*1, *0\} \approx *2$

$H_4 = \{H_3, H_1 + H_2\} \approx \{*2, *0 + *1\} \approx *0$

Grundy sequence: [0,0,1,2,0,1,2,3,1,2,3,4,0,3,4,2,1,3,2,1,0,2,1,4,5,
1,4,5,1,2,0,1,2,3,1,2,3,4,2,3,4,2,3,4,2,1,0,2,8,4,5,3,4,5,6,2,5,1,2,3,1, 2,3,4,2,3,4,2,3,4,2,3,0,2,3,4,5,3,4,5,6,4,
3,0,2,3,4,5,3,4,5,6,4,5,6,8,5,1,2,3,4,2,3,4,2,3,4,2,3,0,2,3,4,1,3,4,5,6, 4,5,6,8,5,6,8,7,2,8,3,4,2,3,4,2,3,0,2,9,3,
3,4,5,3,4,2,3,0,2,9,3,5,4,8,5,4,8,5,4,8,5,6,8,9,2,8,9,6,5,3,4,5...

Bounded Grundy Values?

Problem: do all octal games have bounded Grundy numbers?

equivalent: is [grundy({1}), grundy({2}), ...] ultimately periodic?

Beware! This is a hard problem. There are octal games with

- huge period: 0.16 has length 149459
- huge pre-period: 0.376 has length 2268247 (and period 4!)
- unknown behaviour: 0.6, 0.06, 0.14, 0.36, 0.64, 0.74, 0.76, 0.004, 0.005, 0.006, 0.007, ... $\text{grundy}_{0.007}(1683655) = 1314$

all data from Guy: *Unsolved Problems in Combinatorial Games*,

in: `http://www.msri.org/publications/books/Book29/contents.html`

see also Flammenkamp: `http://wwwhomes.uni-bielefeld.de/achim/octal.html`

Grundy Classes for Rewrite Games

Proposition: for each octal game G:

Grundy values are bounded $\iff$ Grundy sequence is ultimately periodic $\iff$ corresponding rewrite game has finitely many non-empty Grundy classes, all of which are rational.

Generalization:

- input: a rewrite system R on signature Σ
- question: is there a bound c such that $\forall t : \text{grundy}_R(t) \le c$?
- question': are $C_k := \{t : \text{grundy}_R(t) = k\}$ rational languages?

Find families of R that fulfill these properties ...
... and that contain (some, or even all) R_{octal}!

Grundy Classes and Set Constraints

$R(L)$ = immediate successors = $\{y : L \ni x \to_R y\}$

$R^-(L)$ = immediate predecessors = $\{x : x \to_R y \in L\}$

for P/N classes: $N = \text{Term}(\Sigma) \setminus R^-(N)$.

for Grundy classes: $C_k = (R^-(C_0) \cap \ldots \cap R^-(C_{k-1})) \setminus R^-(C_k)$

If R terminates $\Rightarrow$ solution $[C_0, C_1, \ldots]$ is unique.

Problem: properties of $R \xrightarrow{?}$ properties of solution.

Study "set constriants with *rewrite* projections".

Examples: $R_{0.6} = \{11 \to 10\}, R_{0.007} = \{111 \to 0\}$.

Grundy Classes for Subtraction Games

"take-and-break" without breaking. octal code digits $\in \{0, 3\}$

rewrite system: $R = \{1^k \to \epsilon : k \in K\}$

known: for each subtraction game,
the Grundy sequence is eventually periodic.

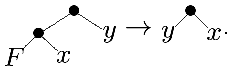
beware: huge periods $K = \{22, 34, 53, 86\} \Rightarrow 114.109$, intricate structure.
investigated by Althöfer, Flammenkamp, Sillke (199*)

alternate proof in our framework:

For one-letter languages $L \in 1^*$, predecessors $R^-(L) = 1^K \cdot L$.

$\Rightarrow$ classical set constraint system $\Rightarrow$ solutions are rational.

Linear inverse monadic systems

Example: $R =$ 

Proposition: If R is terminating, linear, non-overlapping, and inverse monadic, then Grundy classes are algebraic tree languages.

Proof: R is strongly confluent (all reductions $t \to^* \mathsf{Nf}(t)$ have common length) $\Rightarrow$ there are only two Grundy classes $\Rightarrow$ membership depends on length of reduction $\Rightarrow$ can be counted.

These Grundy classes are not necessarily rational. (Example.)

James Bond $\{111 \to 0\} \cong \{1(1(1(x))) \to 0(x)\}$ *is* linear inverse monadic, but overlapping.

Challenge Problems

basic: For the Autobahn game $= \{01 \to 10, 1 \to 2\}$
(think: 0 = empty space, 1 = car, 2 = car wreck),
prove that each $(01010101)^k$ is a second-player win.

report solution to me, `joe@informatik.uni-leipzig.de`*,*
and collect $10 prize money

medium: For the James Bond game $= \{111 \to 0\}$, determine whether there are infinitely many second-player wins among 1^*.

advanced: Prove that each octal game
has a bounded Grundy sequence.

report solution to Berlekamp/Conway/Guy, and become world famous . . .