Re: games and multiple quantifiers

Source: http://sci.tech−archive.net/Archive/sci.math/2005−01/4445.html

---

From: KRamsay (*kramsay_at_aol.com*)
Date: 01/15/05

```
Date: 15 Jan 2005 08:18:09 GMT
```

In article <wcSFd.39371$8m.751177@weber.videotron.net>, David Bernier
<david250@videotron.ca> writes:
|To express that White has a winning strategy in <= 60 moves in chess,
|(White: 60 moves, Black: 60 moves), one could write:
|
|(E w1) (A b1) (E w2) (A b2) .... (E w60) (A b60)
| [WhiteWins(w1, b1, w2, b2, ... w60, b60] .
|
|Here, WhiteWins(.) can be taken to be a recursive function
|on 120 integers restricted to a domain D^120 where
|D = {1, 2, ... NMAX}, and NMAX is not too large,
|certainly NMAX < 10000.

Just as an aside here about the finite case....

This is a type of problem that converts naturally into instances
of "quantified boolean formula", where the variables are restricted
to just boolean variables. Quantified boolean formula is the classic
example of a PSPACE (computable in polynomial−bounded space)
complete problem.

|If f is a recursive function on N^120,

I assume by "N" you mean the natural numbers.

|is it the case that even having an
|oracle for the halting problem,
|
|it need not be the case that:
|(E w1) (A b1) (E w2) (A b2) .... (E w60) (A b60)
| [f(w1, b1, w2, b2, ... w60, b60]
| is decidable?

The terminology here is a little delicate. What you have
written is only a statement if the output of f is taken
somehow to represent "true" or "false". Perhaps f returns
1 and 0 for those. If so, then you have a single statement
here. "Decidable" as an adjective applied to an individual

statement seems not to be used so much, and it's used
in relation to a theory: it means either provable or disprovable
in the given theory.

Usually decidable refers to a class of statements. To make
this statement a class we would replace f with a parameterized
sequence of recursive relations. Actually, the usual thing
is to consider what are called "primitive recursive" predicates.
The primitive recursive functions are computable by programs
that have no "unbounded" looping constructs. They can be
enumerated.

If f(k; w1, b1, ..., w60, b60) is an enumeration of the 120−place
primitive recursive predicates, the problem of determing for a
given k whether

(Ew1)(Ab1)... f(k; w1,b1,...,w60,b60)

is undecidable, i.e., there's no algorithm for it. Denote by
S the set of such k's.

There's a hierarchy of "logical complexity" here. The primitive
recursive predicates themselves are counted as both pi−0−0
and sigma−0−0. The predicates expressible with one existential
quantifier are sigma−0−1. The ones expressible with one universal
quantifier are pi−0−1. The set of k's that make the statement above
are sigma−0−120. As you go up the hierarchy, each level properly
contains the ones below it; i.e., every sigma−0−n and every pi−0−n
predicate is both sigma−0−(n+1) and pi−0−(n+1) too by just adding
a dummy parameter, but there are sigma−0−(n+1) predicates and
pi−0−(n+1) predicates that are neither sigma−0−n nor pi−0−n. This can
be proven using clever adjustments to familiar diagonal arguments.

The set S is a sigma−0−120 complete set, meaning that any other
sigma−0−120 set can be reduced to it. (We have no problem with
the different senses of complete here: the conversion is of a low
complexity.)

If a problem is decidable by an oracle Turing machine, with an
oracle to the halting problem, then it is both sigma−0−2 and pi−0−2
(which is denoted "delta−0−2"). That for a given input the oracle
Turing machine answers "yes" can be encoded by saying that
there exists an integer N1 such that every integer N2 has the
following property. N1 encodes a possible path for the oracle
Turing machine to make together with halting computations for
all the places in the computation where the oracle answers "yes"
to an instance of the halting problem, and N2 does not encode
a halting computation for any of the instances where the oracle
answered "no".

On the other hand, that the oracle Turing machine answers "no" in a given instance can also be expressed in the same form. That means the fact that it answers "yes" in a given instance can be expressed as the negation: for every N1 there exists an N2 satisfying a property (which one can check is primitive recursive). So it can be expressed both in sigma−0−2 and pi−0−2 form (i.e., is delta−0−2) as I claimed.

Well, delta−0−2 is way below your sigma−0−120 complete set S, so there's no hope that S is computable by such an oracle machine.

Keith Ramsay