Death, Taxes, and Formal Verification

Justin Hsu

University of Pennsylvania
3330 Walnut Street, Philadelphia, USA
justhsu@cis.upenn.edu

---

Abstract

Tax avoidance is estimated to cost the United States $184 billion every year [2]. A substantial portion of this shortfall comes from loopholes, corner cases in the tax code that are exploited by sophisticated tax strategies. Closing these loopholes is frequently touted as an easy, uncontroversial way to boost revenue. While the tax code is frequently updated, tax experts continue to find creative ways to evade taxes; it seems that closing tax loopholes may not be so easy after all. To give an idea of the challenges, let’s consider recent efforts by the American Internal Revenue Service (IRS) to prohibit the Killer B transaction, named after a section of the tax code.

First, a bit of background. The US corporate tax rate is substantially higher than the rate in tax havens like Ireland, and corporations are charged tax on profits worldwide. As a result, companies have a strong incentive to transfer ownership overseas—known as inversion. However, the US tax code requires foreign assets to be taxed at the US rate if they are transferred back to the US. As a result, companies frequently have foreign assets that they cannot easily access.

The Killer B maneuver is designed to solve this problem. Suppose a US parent company P owns a foreign subsidiary FS and a domestic subsidiary DS. Initially, DS itself owns a foreign subsidiary FT, the target. Suppose P wants to repatriate assets (say, cash) held abroad by FS. Under a so-called triangular reorganization, FS pays cash to P in exchange for shares of P. Then, FS immediately swaps the shares of P to DS in exchange for T. By a combination of tax codes, the whole transaction is tax free, and P ends up with the foreign assets home free.

The IRS cracked down with IRS notice 2006-41. However, IBM found a new loophole. Instead of having FS buy stock from P, they had FS buy shares of P on the foreign stock exchange. As IBM was interested in buying back shares, this was perfect: they had effectively bought back shares using foreign assets. This loophole was closed with IRS notice 2007-48 two days later [3].

In 2014, tax advisers working on the merger of Liberty Global (a US company) and Virgin Media (a UK company) found yet another loophole when trying to invert Liberty Global to a foreign company. Under the earlier, “Helen of Troy” rules, inverting companies are taxed at a high rate. However, there was a rule that prevented the Helen of Troy rules and the Anti-Killer B rules from both firing: If the amount “recognized” for taxation under the latter rules was higher than under the former rules, only the latter rules would apply. The merger recognized the tax under the Anti-Killer B rule, so the Helen of Troy rules were turned off. However, the tax was recognized in the UK, where there was no tax liability. The result: a tax-free inversion [1].

As these examples illustrate, a serious obstacle in closing loopholes is the complexity of the tax code. Patches to prevent certain maneuvers can enable other holes, kind of like patching computer bugs. In some sense, the tax code is a very complex program, calculating the tax amount based on the individual’s or corporation’s actions.

We believe that applying formal methods to this problem can help recognize and patch loopholes in a more rigorous way. While the tax code is very complex, it attempts to be rigorous. For instance, there are definitions for what a corporation is, and what operations a corporation may perform. From a formal verification point of view, the problem of preventing repatriation without taxation seems like a problem of provenance, or of secure information flow: The goal is to control how information flows from one place (abroad) to another (back home). Whether or not the complex tax code (or perhaps a core version) can be effectively formalized is so far unknown, but the potential gains are large and the motivation is strong.
References

1. IRS aims at innovative M&A inversion structure.