Property Law as a Programming Language

Shrutarshi Basu
Computer Science
Cornell University

James Grimmelmann
Cornell Law School

Nate Foster
Computer Science
Cornell University

Abstract

Anglo-American law enables property owners to split up rights among multiple entities by breaking their ownership apart into *future interests* that change over time. The *conveyances* that owners use to transfer and subdivide property rights follow rigid syntactic conventions and are governed by an intricate body of interlocking legal doctrines determining their legal effect over time. These doctrines have been codified, but only in informal and potentially ambiguous ways. This paper presents preliminary work in developing a formal model for expressing and analyzing property conveyances. We show that the syntactic and semantic structure of conveyances are amenable to modeling and analysis using language-based techniques.

1 Introduction

Anglo-American property law governs what things can be owned, what rights owners enjoy, and how these rights can be transferred and divided over time. Some portions of property law, such as nuisance law (governing relations among neighors) are based on open-ended standards and interpretation via common sense balancing tests. But other portions are notorious for their rigidity, relying on intricate bright-line rules that leave little room for interpretation.

The most rigid and intricate portion of property law is the system of doctrines governing the division of ownership of property over time. Although the system can be used to express great flexibility in practice, this flexibliity is achieved by way of a dense tangle of doctrines that govern the permissible *interests* in a piece of property through legal statements called *conveyances*. For example, one person might give ownership to a tract of land to another person for their lifetime, then to a second person until some condition is fulfilled, and finally to a third person permanently thereafter, creating a chain of three interests.

This paper demonstrates initial work towards a formal treatment of interests in property law. We treat conveyances as programs in a DSL that encode a variety of standard legal concepts. Such a program denotes a deterministic tree of property interests. By applying a series of events to an interest tree, we can determine the who possesses the rights to a piece of property at any given point in time.

DSLDI'17, October 22, Vancouver, Canada
2018. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

O conveys	to A.	(1)
O conveys	to A for life, then to B.	(2)
O conveys	to A for life while A lives on the property, then to B.	(3)
O conveys	to A for life, but if B gets married then to B.	(4)
O conveys	to A for life, then to B for life if B has turned 21, then to C.	(5)

Figure 1. Some example conveyances

2 Examples and Terminology

Property law deals with the rights of *owners* of property. For real estate, these rights are termed *present estates* if their owner can currently excercise them, or *future interests* if they can only be exercised at some point in the future (if at all). Interests are created (and the corresponding rights transferred) from one party (the *grantor*) to another (the *grantee*) in a variety of ways (wills, deeds, etc.), which we will collectively call *conveyances*.

Figure 1 shows a number of example conveyances. A conveyance starts with a *grantor* (in the examples, *O*), followed by a number of *clauses* separated by commas. Clauses can take several different forms, which determine the kinds of interests they create. For example, Conveyance 1 has a single clause: *O* conveying her entire interest to *A*, who is entitled to possess the property now and forever, and also to convey it to another grantee if he so desires.

In Conveyance 2, grantor *O* conveys an interest to a grantee *A* for a limited time: *A*'s lifetime. After that, *B* is to receive the property. In this case, *A*'s interest is deemed to have a *natural duration*—*A*'s lifetime. Conveyance 3 attaches an *added limitation* to *A*'s interest: *A* ceasing to live on the property.

Conveyance 4 starts to show some of the subtleties of interpreting conveyances from a legal perspective: *A*'s interest may be terminated either by *A*'s death or *B*'s marriage. However, syntactically, the condition "*B* gets married" is attached to the clause determining *B*'s interest (it is written after the last comma in the conveyance). In legal terms, *B*'s marriage is an *executory limitation*—it is a precondition on *B*'s interest that terminates all preceding interests.

Finally, Conveyance 5 shows yet another syntactic quirk in the structure of conveyances. On the surface, it is similar to Conveyance 4: the condition of B turning 21 is attached to B's interest. However, since the condition uses the wording "then to B if" rather than "but if ... then to B" it is considered a *condition precedent* that must be satisfied before B takes possession, but does not automatically end prior interests.

These examples show that while conveyances have a simple syntactic structure, the precise placement of conditions and keywords in the text have a dramatic impact on how interests are created and terminated. This is an application ripe for a DSL: a small, custom language that cleanly expresses the complexity of conveyances, and whose semantics closely match the legal interpretation.

Strings	s	::=	Strings	
Persons	p	::=	$o, g, a, b \ldots$ Owner, grantor & others	
Events	e	::=	p dies | n years pass	
		|	p re-enters | s occurs	
Predicates	π, ϕ	::=	Occurs e	
		|	$\neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$	
		|	Previously p | Always p	
		|	Sometime p | p Until q	
		|	true | false	
Preconditions	ψ	::=	$CP(\phi)$	Cond. Precedent
		|	$CS(\phi)$	Cond. Subsequent
		|	$EL(\phi)$	Exec. Limitation
		|	$AC(\phi)$	Alt. Contingents
		|	$\varnothing$	Unconditional
Durations	δ	::=	Life(p)	Life Estate
		|	Term(n)	Term of Years
		|	Abs	Absolute
Limitation	λ	::=	ϕ	Added Limitation
Interest	i	::=	$(\psi, p, \delta, \lambda) \mid i; i$	
Conveyance	c	::=	$(g, i) \mid c; c$	Sequencing
Interest graph	γ	::=	Atom (p)	Atomic Interest
		|	Until $(\phi, \gamma_1, \gamma_2)$	Sequencing
		|	If $(\phi, \gamma_1, \gamma_2)$	Branching

Figure 2. Abstract Syntax.

3 Formalizing Conveyances and Interests

Figure 2 shows our current abstract syntax for conveyances. It is organized around persons p, events e, and predicates ϕ or π. The predicate language is based on linear temporal logic—i.e., standard Boolean and temporal operators for defining relationships between events in an ordered history (eg., And, Or, Sometime, Until, etc.). An interest is simply a 4-tuple: a *precondition* ψ, an owner p, a natural durations δ, and an added limitation λ. Preconditions include conditions precedent and subsequent, and executory limitations, allowing us to express the concepts shown in the examples. A conveyance combines a grantor g with a list of interests i, and allows sequencing. In our experience, this syntax models a wide range of conveyances, but more constructs may need to be added or refined in the future.

The abstract syntax thus far is a simpler representation of legal conveyances, but it is still a little unwieldy for computation. For example, both natural duration and limitations involve the same concept: conditions under which an interest expires. An interest graph γ abstracts out everything except the owners of interests and the conditions under which possession passes from one interest to the next. An interest graph is built by three constructors: *Atom* to create individual interests for each grantee, *Until* $\phi, \gamma_1, \gamma_2)$ representing the termination of interests in γ_1 on ϕ and finally *If*$(\phi, \gamma_1, \gamma_2)$ representing transfer of ownership to γ_1 on ϕ and to γ_2 on $\neg\phi$. By allowing these elements to be nested, we can construct complex terms that precisely describe the flow of possession between interests once the associated conditions are fulfilled.

Abstract conveyances can be derived from the corresponding natural language using a straightforward parser. Interest graphs γ are derived from conveyances c via a denotational translation, which we elide due to space constraints.

By boiling down the complex language of conveyances into combinations of predicates and atomic interests, we can leverage programming language tools such as temporal logics and finite state automata. We are currently exploring one such avenue: constructing useful finite state automata from interest graphs. To do this, we map atomic interests to individual states, and events to transitions between those states. Borrowing from temporal logics, we map predicates to paths (i.e., sequences of events) through the automaton. Such an automaton consumes a sequence of events (a past history), and by inspecting the resulting current and reachable future states, we can tell the current and possible future ownerships of the property in question.

4 Applications and Implementation

Our model allows us to answer interesting questions regarding interests and possession. For example, after feeding a sequence of events to an automaton, the possessor is simply the person associated with the current state. Furthermore, any states that are now unreachable from the current state denote interests that have expired, or are no longer capable of taking possession due to the events that have transpired.

We have built a system in OCaml that converts written conveyances in a human-readable form to a visual representation of an interest graph. We can apply a sequence of events to the graph, which then shows the current and possible future states. The system is accessible via a web interface and is being tested with examples from legal casebooks.