Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Topic 1

Digital Rights Licensing

What is Digital Rights?

- Digital rights deal with the rights of owners of artistic expressions
  - ⋆ music, movies, readings, photographs, paintings, ...
- in the context of the downloading of these
- via the Internet to users
- who are then supposed to pay for the right to render,
- i.e., to listen, see, hear, and see–see, ..., these.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Realities by Users and Licenses Issued by Owners

- Users perform **pay**ment and **render**ing **events**.
- Sequences of events as performed by users make up **realities**.
- Owners issue **licenses** which describe which realities are permissible.

Digital Rights Management (DRM)

- DRM is then about
  - ⋆ the **design** of appropriate **license languages**
  - ⋆ and the **enforcement** of user realities
  - ⋆ in order for these to **not breach**, but to **fullfil** the licenses.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Structure of Presentation

- First a loyal, but problematics transcription of a published paper.
- Then a “similar”, but believed correct reformulation (in `RSL`).

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Topic 2

Transcript of the Gunter/Weeks/Wright Paper

Actions, Events, Realities and Licenses

- The model centers around
  - ⋆ a domain of *realities* and
  - ⋆ a domain of *licenses*,
- where
  - ⋆ A *reality* is a sequence of events.
  - ⋆ A *license* is a set of realities.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- The semantics of a rights management languages can be expressed as a function
- that maps terms of the language to elements of their domain of licenses.
- Their abstract model
  - ⋆ represents an *event*, $e \in Event$, as a pair of a *time*, $t \in Time$,
  - ⋆ and an *action*, $a \in Action$:

$$e ::= t : a$$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- $Time$ is totally ordered by $<$.
- The function $+$:
  - ⋆ $Time \times Period \rightarrow Time$
  - ⋆ adds a period, $p \in Period$, to a time.
- There are two kinds of actions:

$$a ::= \textsf{render}[w, d] \mid \textsf{pay}[x]$$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- $w \in Work$ denotes a rights-managed work.
- $d \in Device$ represents a DRM-enabled device.
- x is a decimal.
- Action $\textsf{render}[w, d]$ represents rendering of work w by rights-enabled device d.
- Action $\textsf{pay}[x]$ represents a payment of amount x of some currency from a licensee to a license issuer.
- The event $t : \textsf{render}[w, d]$ means that at time t, work w was rendered on device d.
- The event $t : \textsf{pay}[x]$ means that at time t, a payment of amount x was made.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- A *reality*, $r \in Reality$,
  - ⋆ is a finite set of events,
  - ⋆ such that all events occur at distinct times:

$$\text{Reality} = \{ \text{ E} \mid \text{E:Event-}\textbf{set} \cdot \text{t:a} \in \text{E} \wedge \text{t:a}' \in \text{E} \Rightarrow \text{a=a}' \}$$

where

- ⋆ $P(x)$ represents the powerset of E (the set of all subsets of E).
- ⋆ Notation:
  - ◇ $r_{\leq t}$ represents the prefix of r that occurs at or before time t;
  - ◇ that is: $r_{\leq t} = \{t' : a \in r \mid t' \leq t\}$.
- ⋆ Notation:
  - ◇ $r \sqsubseteq r'$ indicates that r is a prefix of r';
  - ◇ that is, there exists a t such that $r = r'_{\leq t}$.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

- In the model, a *license*, $l \in License$, is a set of realities:

$$l \in License = P(Reality)$$

- Let us take an example:

$$l_A = \left\{ \begin{array}{l} \{8:00:\mathsf{pay}[\$1],\ 8:01:\mathsf{render}[w_1,d_1]\}, \\ \{8:00:\mathsf{pay}[\$1],\ 8:02:\mathsf{render}[w_1,d_1]\}, \\ \{8:00:\mathsf{pay}[\$1],\ 8:03:\mathsf{render}[w_1,d_1], \\ \qquad\qquad\qquad 8:04:\mathsf{render}[w_1,d_1]\}, \\ \{8:00:\mathsf{pay}[\$1]\} \end{array} \right\}$$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

- To give a more formal meaning to a license,
- suppose $\mathbf{r}$ is the (unique) complete reality
- that actually occurs over the entire lifetime of the DRM system.
- Let $\mathbf{r}[l]$ be events of $\mathbf{r}$ attributed to license l.
- **Definitions**
  - ⋆ Reality $r \in l$ of license l is *viable* for $\mathbf{r}[l]$ at t iff $\mathbf{r}[l]_{\leq t} \sqsubseteq r$.
  - ⋆ License l is *fulfilled* by $\mathbf{r}[l]$ at t iff $\mathbf{r}[l]_{\leq t} \in l$.
  - ⋆ License l is *breached* by $\mathbf{r}[l]$ at t iff there does not exist $r \in l$ that is viable for $\mathbf{r}[l]$ at t.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

- **Example 1:**

  $\mathbf{r}[l_A] = \{$ 8:00:$\mathsf{pay}[\$1]$, 8:01:$\mathsf{render}[w_1, d_1]$, 8:05:$\mathsf{render}[w_1, d_1]$ $\}$.

  For $\mathbf{r}[l_A]$,
  - ⋆ at $t <$ 8:01, all four realities of license l_A are viable
  - ⋆ at t, 8:01 $\leq t <$ 8:05, only the first reality is viable
  - ⋆ at $t \geq$ 8:05, no reality is viable

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

- **Example 2:**

  $\mathbf{r}[l_A] = \{$ 8:00:$\mathsf{pay}[\$1]$, 8:01:$\mathsf{render}[w_1, d_1]$, 8:05:$\mathsf{render}[w_1, d_1]$ $\}$.

  License l_A is
  - ⋆ unfulfilled by $\mathbf{r}[l_A]$ for $t <$ 8:00
  - ⋆ fulfilled for 8:00 $\leq t <$ 8:05
  - ⋆ breached for $t \geq$ 8:05

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- **Example 3:**
  r'$[l_A] = \{$ 8:00:pay$[\$1]$, 8:03:render$[w_1, d_1]$ $\}$.

  license l_A is

  ⋆ unfulfilled for $t <$ 8:00

  ⋆ fulfilled for 8:03 $\leq t <$ 8:03

  ⋆ unfulfilled for 8:03 $\leq t <$ 8:04

  ⋆ breached for $t \geq$ 8:04

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Standard Licenses
Up Front Licenses

- The "UP Front" license provides access
- to any work in set $W \in P(Work)$
- on any device in set $D \in P(Device)$
- beginning at time t_0 for period p,
- for an up-front payment of x:

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

$$
\begin{aligned}
&\mathrm{UpFront}(t_0, x, p, W, D) = \\
&\quad \{t_0 : \mathsf{pay}[x], \\
&\quad\ \ t_1 : \mathsf{render}[w_1, d_1], ..., t_n : \mathsf{render}[w_n, d_n] \\
&\quad \mid n \geq 0, \\
&\quad\ \ t_0 < t_1 < ... < t_n < t_0 + p, \\
&\quad\ \ w_1, ..., w_n \in W, d_1, ..., d_n \in D\ \}
\end{aligned}
$$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Problems

- The use of n is confusing.
- On one hand it is used to express up to n renderings.
- On the other hand subscript $_n$ is used for ranging both works and devices.
- The three uses of n should be separated into, say, i, j and k.
- The same comments apply to the Flat Rate and Per Use formulations.
- Why not allow $t_n \leq t_0 + p$?

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Flat Rate Licenses

- The "Flat Rate" license provides access
- to any work in set W
- on any device in set D
- beginning at time t_0 for period p,
- for a payment of x at the end of the period:

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

$$
\begin{aligned}
&\mathrm{FlatRate}(t_0, x, p, W, D) = \\
&\quad \{t_1 : \mathsf{render}[w_1, d_1], \ldots, t_n : \mathsf{render}[w_n, d_n] \\
&\quad\ \ t_{n+1} : \mathsf{pay}[x], \\
&\quad \mid n \geq 0, \\
&\quad\ \ t_0 < t_1 < \ldots < t_n < t_{n+1} < t_0 + p, \\
&\quad\ \ w_1, \ldots, w_n \in W, d_1, \ldots, d_n \in D \ \}
\end{aligned}
$$

Problems

- Why not allow $t_n \leq t_0 + p$?

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Per Use Licenses

- The "Per Use" license is provides access
- to any work in set W
- on any device in set D
- beginning at time t_0
- for a period of length p,
- for a payment of x per use at the end of the period:

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

$$
\begin{aligned}
&\mathrm{PerUse}(t_0, x, p, W, D) = \\
&\quad \{t_1 : \mathsf{render}[w_1, d_1], \ldots, t_n : \mathsf{render}[w_n, d_n] \\
&\quad\ \ t_{n+1} : \mathsf{pay}[nx], \\
&\quad \mid n \geq 0, \\
&\quad\ \ t_0 < t_1 < \ldots < t_n < t_{n+1} < t_0 + p, \\
&\quad\ \ w_1, \ldots, w_n \in W, d_1, \ldots, d_n \in D \ \}
\end{aligned}
$$

Problems

- Why not allow $t_{n+1} \leq t_0 + p$?

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Up to Expiry Date Licenses

- A license that a consumer can accept any time before some future
  - ⋆ can be constructed by quantifying over t_0
  - ⋆ for any of the three families defined on preceding slides.
- For example,

$$\bigcup_{t_0 < t_{\text{expires}}} \text{UpFront}(t_0, x, p, W, D)$$

- This license allows the period of the use to begin anytime prior to t_{expires}.

Problems

- Why not allow $t_0 \leq t_{\text{expires}}$?

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Non-cancellable Multi-use Licenses

- To construct multi-period non-cancellable licenses
- by composing single-period licenses,
- an operator $\triangle$ is defined.

$$l_1 \triangle l_2 = \{r_1 \cup r_2 \mid r_1 \in l_1, r_2 \in l_2\}$$

- here all of the events of l_1 occur at different times from the events of l_2.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Problems

- The above is not the same as: $l_1 \triangle l_2 = l_1 \cup l_2$
- How does $\triangle$ associate?
- We guess: $l \triangle\, l' \triangle\, l'' = l \triangle\, (l' \triangle\, l'')$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

- $\text{UpFront}^{\triangle}(t_0, x, p, W, D, m)$
- provides access to any work in set W
- on any device in set D
- beginning at time t_0
- for m periods of length p,
- for payments of x at the beginning of each period:

$$\begin{aligned}&\text{UpFront}^{\triangle}(t_0, x, p, W, D, m) = \\ &\quad \text{UpFront}(t_0, x, p, W, D) \triangle \ldots \triangle \\ &\quad \text{UpFront}(t_{m-1}, x, p, W, D) \\ &\qquad \text{where } t_i = t_0 + ip \text{ for } i \text{ from } 0 \text{ to } m-1\end{aligned}$$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Cancellable Multi-use Licenses

- With the $\triangleright$ operator cancelable multi-period licenses can be defined.

$$l_1 \triangleright l_2 = \{r_1 \cup r_2 \mid r_1 \in l_1,\ r_1 \neq \emptyset,\ r_2 \in l_2\} \cup \{\emptyset\}$$

- here all of the events of l_1 occur at different times from the events of l_2.

Problems

- Same question concerning associativity.
- What, exactly is the rôle of the empty set $\{\}$, or, rather, $\{\{\}\}$ (in Gunter et al. paper: $\{\emptyset\}$).
- We guess: To have only actions from l_1 and then "get out"!
- Also: There is a probem with t_i: it is defined but never used!

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- $\text{UpFront}^{\triangleright}(t_0, x, p, W, D, m)$
- provides access to any work in set W
- on any device in set D
- beginning at time t_0
- for m periods of length p,
- for payments of x at the beginning of each period,
- cancelable after any period:

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

$$\begin{aligned}&\text{UpFront}^{\triangleright}(t_0, x, p, W, D, m) =\\ &\quad \text{UpFront}(t_0, x, p, W, D) \triangleright \ldots \triangleright\\ &\quad \text{UpFront}(t_{m-1}, x, p, W, D)\\ &\qquad \text{where } t_i = t_0 + ip \text{ for } i \text{ from } 0 \text{ to } m-1\end{aligned}$$

Problems

- Same question concerning associativity.
- There is still a probem with t_i: it is defined but never used!

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

A License Language

Syntax

The language *DigitalRights* is defined by the following grammar:

$$\begin{aligned}e ::= &\ (\mathsf{at}\ t \mid \mathsf{until}\ t)\\ &\ (\mathsf{for}\ p \mid \mathsf{for}\ [\mathsf{up}\ \ \mathsf{to}]\ m\ p)\\ &\ \mathsf{pay}\ x\ (\mathsf{upfront} \mid \mathsf{flatrate} \mid \mathsf{peruse})\\ &\ \mathsf{for}\ W\ \mathsf{on}\ D\end{aligned}$$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Examples

until 01/01/03
for up to 12 months
pay $ 10.00 upfront
for "Jazz Classics"
on "devices registered to license holeder"

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Semantics

$$\mathcal{M}[\text{at } t\ z] = \mathcal{M}_1[z](t)$$
$$\mathcal{M}[\text{until } t\ z] = \bigcup_{t'<t} \mathcal{M}_1[z](t')$$

$$\mathcal{M}_1[\text{for } p\ z](t_0, p) = \mathcal{M}_2[z](t_0, p)$$
$$\mathcal{M}_1[\text{for up to } m\ p\ z](t_0) = \mathcal{M}_2[z](t_0, p) \rhd \ldots \rhd \mathcal{M}_2[z](t_{m-1}, p)$$
$$\text{where } t_i = t_0 + ip \text{ for } i \text{ from } 0\ to\ m-1$$
$$\mathcal{M}_1[\text{for } m\ p\ z](t_0) = \mathcal{M}_2[z](t_0, p) \triangle \ldots \triangle \mathcal{M}_2[z](t_{m-1}, p)$$
$$\text{where } t_i = t_0 + ip \text{ for } i \text{ from } 0\ to\ m-1$$

$$\mathcal{M}_2[\text{pay } x \text{ upfront for } W \text{ on } D](t, p) = \text{UpFront}(t, x, p, W, D)$$
$$\mathcal{M}_2[\text{pay } x \text{ flatrate for } W \text{ on } D](t, p) = \text{FlatRate}(t, x, p, W, D)$$
$$\mathcal{M}_2[\text{pay } x \text{ peruse for } W \text{ on } D](t, p) = \text{PerUse}(t, x, p, W, D)$$

Problems

- Parameters (t_0, p) in first lefthand side of $\mathcal{M}_1$ is wrong,
- should be just (t_0).

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Topic 3
An RSL Model
Actions, Events, Realities and Licences

type
T, W, D, P
E = T × A
A == mkR(w:W,d:D) | mkP(x:P) or: **rndr**(w,d) | **pay**(x)
R = {| evs:E-**set** · wfEvs(evs) |}

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Annotations

- T, W, D, and P stands for time, work, device and payment.
- Events, e:E, are Cartesian pairs of times (i.e., time stamps) and actions.
- Actions are discrimated, either renderings (which are records mkR(w,d)) of works and devices. or payments (mkP(x)) of currency amounts.
- The trailing "or" shows our schematic way of representing actions.
- Realities, r:R, are well-formed sets of events.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

value

wfEvs: E-**set** → **Bool**

wfEvs(evs) ≡ ∀ (t,a),(t′,a′):E•{(t,a),(t′,a′)}⊆evs ∧ t=t′ ⇒ a=a′

$r_{\leq t}$: prefix: R → T → R

$r_{\leq t}$ ≡ prefix(r)(t) ≡ {(t′,a)|(t′,a):E • (t′,a) ∈ r ∧ t′≤t}

$r' \sqsubseteq r$: is_prefix: R × R → **Bool**

$r' \sqsubseteq r$ ≡ is_prefix(r′,r) ≡ ∃ t:T • r′ = prefix(r)(t)

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Annotations

- A set of events form a reality if no two events have the same time stamp.
- The prefix of a reality up to and including time t is the set of all those events of the reality whose time stamp is equal to or less than t.
- A reality is a prefix of another event if there is a time t such that the former reality is a prefix of the latter reality.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

type

L = R-**set**

value

/∗ viable: capable of working ∗/

is_viable: R → R → L → T → **Bool**

is_viable(r′)(r)(l)(t) ≡ r ∈ l ∧ is_prefix(prefix(r′)(t),r)

/∗ fulfilled: a consumer reality r′ satisfies a license reality ∗/

is_fulfilled: R → L → T → **Bool**

is_fulfilled(r′)(l)(t) ≡ prefix(r′)(t) ∈ l

is_breached: R → L → T → **Bool**

is_breached(r′)(l)(t) ≡ ∼∃ r:R • r ∈ l ∧ is_viable(r′)(r)(l)(t)

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Annotations

- A reality r' is viable at a time t wrt. a reality r of a license l if r' is a prefix, at that time, of r.
- A reality r' is, at time t fulfilled wrt. a license l if the prefix of r' at time t is a reality of the license.
- License l is breached by consumer reality r' if there is no reality r in l that is viable for r' at t.

Standard Licences
Syntax

type
 I
 Std_L = UpFront|FlatRate|PerUse|Until|NonCanMuUpFro|CanMuUpFro
 Basics = T×P×I×W-**set**×D-**set**
 UpFront == mkUF(sb:Basics)
 FlatRate == mkFR(sb:Basics)
 PerUse == mkPU(sb:Basics)
 Until == mkU(sb:Basics,st:T)
 NonCanMuUpFro == mkNCMF(sb:Basics,sm:**Nat**)
 CanMuUpFro == mkCMF(sb:Basics,sm:**Nat**)

Annotations

- I stands for an interval, i,e,, a time period.
- Std_L stands for standard licenses.
- There are six forms of standard licences: UpFront, FlatRate, PerUse, Until, NonCanMuUpFro and CanMuUpFro.
- They all share some basic information Basics, a Time limit, Payment amount, Interval, identification of the set of Works covered by the license, and identification of the set of Devises overed by the license.
- The NonCanMuUpFro and CanMuUpFro commands furth specify a natural, usually non-zero number (of times of rendering).
- The Std_ definition defines the set of commands as a union using the | type constructor.

- Each individual type is then defined by distinctly named record type constructors: mkUF, mkFR, mkPU, mkU, mkNCMF and mkCMF.
- We have for ease of recalling these mnemonics chosen to name the conctructors with an initial mk (for 'make') and then an abbreviation of the type name being defined.
- The s...: parts of the body of the record type expressions designate selector functions.
- Meta-linguistically:

type
 A, B, ..., C
 R == mkR(sa:A,sb:B,...,sc:C)
axiom
 ∀ r:R, a:A, b:B, ..., c:C •
  r = mkR(sa(r),sb(r),...,sc(r)) ∧
  a = sa(mkR(a,b,...,c)) ∧
  b = sb(mkR(a,b,...,c)) ∧
  ... ∧
  c = sc(mkR(a,b,...,c))

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Semantics

value
  M: Std_L → L
  M(mkUF(t0,x,p,ws,ds)) ≡
   {{(t0,**pay**(x))} ∪
    let ls = [ti↦**rndr**(w,d)|ti:T,w:W,d:D•ti ∈ ts∧w ∈ ws′∧d ∈ ds′] **in**
    {(ti,ls(ti))|ti:T•ti ∈ **dom** ls} **end**
   |n:**Nat**,ts:T-**set**,ws′:W-**set**,ds′:D-**set**•
   card ts=n∧ws′⊆ws∧ds′⊆ds∧t0<**min**(ts)∧**max**(ts)≤t0+p}

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

Annotations

- The above formulation follows that of Gunter et al.,
- but where their model is plain wrong: it does not designate all combinations of works and devices, ours is right:
  - ⋆ At t_0 payment is issued;
  - ⋆ the above expression[1] has two set comprehensions:
  - ⋆ {{A} ∪ {B|C•D} | E•F }
  - ⋆ The inner comprehension {B|C•D} expresses all possible sequences of n renderings involving any combination of works w and devices d from subsets ws and ds of works anbd devices.

[1] { {(t0,**pay**(x))} ∪ {(ti,**rndr**(w,d)) | ti:T,w:W,d:D • ti∈ts ∧ w∈ws′ ∧ d∈ds′ } | n:**Nat**,ts:T-**set**,ws′:W-**set**,ds′:D-**set** • **card**ts=n ∧ ws′⊆ws ∧ ds′⊆ds ∧t0<**min**(ts) ∧ **max**(ts)≤t0+p }

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

  - ⋆ The outer comprehension "selects" an arbitrary, finite n, a set ts of n time points all of which lies between t_0 and $t_0 + p$, and arbitrary subsets ws and ds or works and devices of those given,
  - ⋆ The inner comprehension ensures that all we express all runs of renderings of length n over all combinations of works and devices.
  - ⋆ The outer comprehension ensures that we express all possible and indefinite length runs.
- A better model would be one which, instead of constructively designating all possible runs, expresses the property that a run is an up front run and all such, for the given arguments, are present.

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292
DTU

type
  PayEvent = T × ({|**pay**|}×**Nat**)
  RndEvent = T × ({|**render**|}×W×D)
  UpFrontLicense = {| ℓs:L • wf_UPL(ℓs) |}
value
  wf_UPL: L → **Bool**
  wf_UPL(ℓ) ≡ ∃ t:T,x:**Nat**,p:P,ws:W-**set**,ds:D-**set** • is_ufl(t0,x,p)(ws,ds,ts)(ℓ)

  is_ufl: (T × **Nat** × P) → (W-**set** × D-**set** × T-**set**) → L → **Bool**
  is_ufl(t0,x,p)(ws,ds,ts)(ℓ) ≡
   ∃ t′:T,x′:X • (t′,(**pay**,x′)) ∈ ℓ⇒ t′=t0 ∧ x′=x ∧
   ∀ ti:T,w:W,d:D • ti ∈ ts ∧ w ∈ ws ∧ d ∈ ds ⇒
     (ti,(**render**,w,d)) ∈ ℓ∧
   ∼∃ (t,(**render**,w′,d′)):RndEvent •
     (t,(**render**,w′,d′)) ∈ ℓ∧ t∉ ts ∧ w′∉ ws ∧ d′∉ ds

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

- A set ℓs of licenses
- is the set of all up front licenses
- designated by M(mkUF(t0,x,p,ws,ds))
- if it satisfies are_all_ufl(t0,x,p)(ws,ds,ts)(ℓs).

value
are_all_ufl: T × **Nat** × P × W-**set** × D-**set** → L-**set** → **Bool**
are_all_ufl(t0,x,p)(ws,ds,ts)(ℓs) ≡
∀ ℓ:L • ℓ∈ ℓs ⇒
∃ n:**Nat**,ws′:W-**set**,ds′:D-**set**,ts:T-**set** •
ws′⊆ws ∧ ds′⊆ds ∧ **card** ts=n ⇒ is_ufl(t0,x,p)(ws′,ds′,ts)(ℓ)

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Annotations

- The "do not exists" clauses shall ensure both largest sets of up front licenses over appropriate time spans, works, and devices and that there is no "junk".
- Otherwise we leave it to the reader to decipher the formulas.

value
M(mkFR(t,x,p,ws,ds)) ≡
{**let** ls = [ti↦**rndr**(w,d)|ti:T,w:W,d:D•ti ∈ ts∧w ∈ ws′∧d ∈ ds′] **in**
{(ti,ls(ti))|ti:T•ti ∈ **dom** ls} **end**
∪{(tn′,**pay**(x))}
|n:**Nat**,tn′:T,ts:T-**set**,ws′:W-**set**,ds′:D-**set**•
card ts=n∧ws′⊆ws∧ds′⊆ds∧t0≤**min**(ts)∧ **max**(ts)<tn′≤t0+p}

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

value
M(mkPU(t,x,p,ws,ds)) ≡
{**let** ls = [ti↦**rndr**(w,d)|ti:T,w:W,d:D•ti ∈ ts∧w ∈ ws′∧d ∈ ds′] **in**
{(ti,ls(ti))|ti:T•ti ∈ **dom** ls} **end**
∪ {(tn′,**pay**(n∗x))}
|n:**Nat**,tn′:T,ts:T-**set**,ws′:W-**set**,ds′:D-**set**•
card ts=n∧ws′⊆ws∧ds′⊆ds∧t0≤**min**(ts)∧ **max**(ts)<tn′≤t0+p}

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

value
M(mkU(te,x,p,ws,ds)) ≡ ∪{M(mkUF(t0,x,p,ws,ds))|t0:T•t0≤te}

M(mkNCMF((t,x,p,ws,ds),m)) ≡ $\mathrm{UpFront}^{\triangle}$((t,x,p,ws,ds),m)

M(mkCMF((t,x,p,ws,ds),m)) ≡ $\mathrm{UpFront}^{\triangleright}$((t,x,p,ws,ds),m)

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

value

Δ: $L^* \to L$
$\Delta(ll) \equiv$
case ll **of**
$\langle l \rangle \to l$,
$\langle l \rangle \hat{}\, ll' \to \{r \cup r' | r,r':R \cdot r \in l \wedge r' \in \Delta(ll')\}$
end

$\rhd$: $L^* \to L$
$\rhd(ll) \equiv$
case ll **of**
$\langle l \rangle \to l$,
$\langle l \rangle \hat{}\, ll' \to \{r, r \cup r' | r,r':R \cdot r \in l \wedge r \neq \{\} \wedge r' \in \rhd(ll')\}$
end

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

value

$\mathrm{UpFront}^{\Delta}$: Basics $\times$ **Nat** $\to$ L
$\mathrm{UpFront}^{\Delta}((t,x,p,ws,ds),m) \equiv$
$\Delta(\langle M(mkUF(ti,x,p,ws,ds)) | i:\mathbf{Nat}, ti:T \cdot i:\{0..m-1\} \wedge ti = t0 + i \times p \rangle)$

$\mathrm{UpFront}^{\rhd}$: Basics $\times$ **Nat** $\to$ L
$\mathrm{UpFront}^{\rhd}((t,x,p,ws,ds),m) \equiv$
$\rhd(\langle M(mkUF(ti,x,p,ws,ds)) | i:\mathbf{Nat}, ti:T \cdot i:\{0..m-1\} \wedge ti = t0 + i \times p \rangle)$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

A License Language

type

DRExpr = Time $\times$ Repetition $\times$ Payment $\times$ WorksDevices
Time == mkAt(t:T) | mkUntil(t:T)
Repetition == mkFor(p:I) | mkRepeat(m:**Nat**,p:I) | mkUpTo(m:**Nat**,p:I)
Payment = P $\times$ Kind
Kind == upfront | flatrate | peruse
WorksDevices = W-**set** $\times$ D-**set**

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

value

$M0(mkAt(t),r,(x,k),wds) \equiv M1(r,(x,k),wds)(t)$
$M0(mkUntil(t),r,(x,k),wds) \equiv \cup\{M1(r,(x,k),wds)(t') | t':T \cdot t' < t\}$

$M1(mkFor(p),(x,k),wds)(t) \equiv M2(x,k)(t,p)$
$M1(mkRepeat(m,p),(x,k),wds)(t) \equiv$
$\rhd(\langle M2((x,k),wds)(ti,p) | i:\mathbf{Nat}, ti:T \cdot i:\{0..m-1\} \wedge ti = t0 + i \times p \rangle)$
$M1(mkUpTo(m,p),(x,k),wds)(t) \equiv$
$\Delta(\langle M2((x,k),wds)(ti,p) | i:\mathbf{Nat}, ti:T \cdot i:\{0..m-1\} \wedge ti = t0 + i \times p \rangle)$

$M2((x,upfront),(ws,ds))(t,p) \equiv M(mkUF(t,x,p,ws,ds))$
$M2((x,upfront),(ws,ds))(t,p) \equiv M(mkFR(t,x,p,ws,ds))$
$M2((x,upfront),(ws,ds))(t,p) \equiv M(mkPU(t,x,p,ws,ds))$

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

Topic 4
Closing for Today
Societal Problems

- Patents versus Licenses
- Punishable versus non-punishable infringement
- What are owners trying to achieve?

Japan Advanced Institute of Science & Technology
School of Information Sciences
1-1, Asahidai, Tatsunokuchi
Nomi, Ishikawa, Japan 923-1292

What is Next?

- On Monday I will review other forms of license languages:
  - ⋆ for a far more general, including fair use of artistic works,
  - ⋆ for public administration, and
  - ⋆ for hospital health care.