Draft, to become part of **The LAC HyperGuide** 18.9.1995
http://vir.liu.se/hg-lac/frameworks/hgrl 1

The HyperGuide Reference Language

Erik Sandewall and Antonio Porto

1 Purpose of the HGRL

The HyperGuide Reference Language (HGRL) is a formal language for expressing scenario descriptions involving actions and change. Its use as a reference language in the hyperguide means that

- the intended models and, thereby, the intended conclusions from a scenario description are defined for the HGRL
- scenario examples are expressed in the HGRL
- various actual approaches are defined in relationship to the HGRL.

— More —

2 Overview

A *chronicle* is a description, using logical formulae, of a number of actions, their general effects, and observations of some properties of the world while the actions are being performed. In simple cases, a chronicle consists of the following kinds of formulae:

action statements, for example $\mathsf{D}([20,25], \textsc{make-coffe})$, saying that the current agent performs the act of making coffee during the period from time 20 to time 25;

observations, for example $\mathsf{H}([25], \textsc{coffee-hot})$, saying that there is hot coffee at time 25;

action laws, for example

$\mathsf{D}(w, \text{MAKE-COFFEE}) \Rrightarrow w \mid \text{COFFEE-HOT} := \mathsf{T}$,

saying that if coffee is made during the period w, then the proposition "the coffee is hot" becomes true during that period.

timing statements, for example $\mathsf{t}_2 < \mathsf{t}_4 - 1$, relating the values of two expressions using temporal constants. Such statements are useful when the timepoint arguments of D and H are constant symbols and not explicit timepoints (for example, t_2 rather than 25).

The letter D can be read as *Do* and H as *Holds* (or *Homogenous*). To accommodate multi-valued properties, the second argument of H can be formed using the function :, for example as in

$\mathsf{H}(\texttt{[}s, t\texttt{]}, f : x)$,

which is intended to say that the "feature" f has the value x during the interval from s to t.

In more complex cases, chronicles may contain observations that refer to multiple timepoints, action statements that refer to conditional or composite actions, and other ontologies for time such as branching time. Also, action symbols and features/propositions may have objects as arguments, for example

$\mathsf{H}(25, \mathsf{stateof}(\text{TRALITE4}) : \texttt{Green})$.

expressing that the state of traffic light number 4 is `Green` at time 25. Logics whereby one can express and reason about chronicles are called *logics of actions and change.*

The purpose of this article is to define the *syntax* of chronicles. An operational semantics for chronicles will be defined successively, beginning with a restricted case, and gradually extending it. Those semantics are presented in successive later articles (links are found on the header page).

3 Types and expressions

The following six types are used:

Periods, representing an interval on the timeline;

Times, representing a point on the timeline;

Objects, representing things in the world where actions are performed and changes take place;

Propositions, representing conditions that may be true or false in that world;

Features, representing an aspect of an object, a combination of objects, or the world itself;

Properties, which are the possible values of features.

The following are some of the ways of combining these types.

1. If s and t are two times, then $[s, t]$ is the period having s as its startingpoint and t as its endingpoint (termination).
2. If w is a period and p is a proposition, then $\mathsf{H}(w, p)$ is the statement that p holds throughout the period w.
3. If f is a feature and x is a property, then $f\!:\!x$ is the proposition that f is bound to x.
4. Features and propositions can take objects and properties as arguments.

Examples. One may e.g. use the HGRL so that

- $\mathsf{H}(w, \textsc{raining})$ means that it is raining throughout the period w.
- $\mathsf{H}([40, 54], \textsc{raining})$ means that it is raining throughout the period from time 40 to time 54.
- $\mathsf{H}(w, \mathsf{hungry}(\textsc{fred}))$ means that the proposition "Fred is hungry" holds throughout the period w.
- $\mathsf{H}(w, \textsc{season}\!:\!\mathtt{autumn})$ means that the feature "(current) season" is bound to the property "autumn" throughout the period w.

- $\mathsf{H}(w, \mathsf{haircolor}(\textsc{mary}) : \mathtt{red})$ means that the feature "the hair color of Mary" is bound to the property "red" throughout the period w.

Regardless of type, the HGRL uses the following kinds of elementary symbols:

- *Variables*, represented by *italic* letters.
- (Ordinary) *constants*, represented by ordinary sans-serif letters.
- *Names*, represented by fonts according to type, as follows:
  - times: sans-serif letters, or numbers.
  - periods: sans-serif letters.
  - objects: SMALL CAPITAL letters.
  - propositions: SMALL CAPITAL letters.
  - features: SMALL CAPITAL letters.
  - properties: `teletype` letters.
- *Functions*, represented by ordinary sans-serif letters.

The terms "variable" and "constant" are here used in the sense of those terms that is normal in logic. By "names" we mean constant symbols that stand in one-to-one correspondence with the object domain of the type in question. Thus, each member of the object domain has a name, and two different names can not denote the same object.

3.1 Application specific symbols

Each application uses its own reportoire of some of the types, namely:

- Names of objects, propositions, features, and properties.
- Functions whose values are objects, propositions, features, or properties. Some applications and some of the work use only proposition-valued and feature-valued functions.

In the examples above, RAINING is a proposition name, FRED is an object name, hungry is a function name from objects to propositions, autumn is a property name, haircolor is a function name from objects to features, and red is a property name.

A vocabulary ν is a structure that specifies arity, argument type(s), and value type for each application specific symbol. The precise definition of vocabularies is in [not yet].

3.2 Application independent symbols

Beyond the application-specific vocabularies, there is a number of predicates, functions, and abbreviations which are used across applications, and which are intrinsic to HGRL as such. They are as follows.

The equality relation $=$ applies within each of the types, but not between types (types are assumed to be disjoint).

The predicates H, D, and X all take two arguments, where the first argument is a period and the second argument is a proposition. In the case of X, the second argument may also be a feature.

The relation $<$ takes two times as arguments. The function $[s,t]$ takes two times as arguments, and has a period as value, representing the period that starts in s and ends in t. The precise semantics of times and intervals is defined below.

The function $:$ takes a feature and a property as arguments, and has a proposition as value. Both $[s,t]$ and $:$ are Herbrand functions, that is, different argument combinations can never result in the same value from the function. Formally,

$$[s,t] = [s',t'] \rightarrow s = s' \wedge t = t',$$

and similarly for $f\!:\!x$.

3.3 Logic formulae

Logic formulae are formed as usual in first-order logic, by composition using the operators $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$, $\forall$, and $\exists$. The symbols T and F are used for the logical constants "true" and "false", respectively. The exhaustive definition of the syntax of the HGRL language is found in [annex notyet].

4 Abbreviations

The following abbreviations are introduced.

- If the first argument of H is formed using the interval operator, then the parentheses for the arguments may be omitted. For example, $\mathsf{H}(\,[s,t], p)$ may be abbreviated as $\mathsf{H}\,[s,t]\,p$. The same applies for D and X.

- $\mathsf{H}(\,[t,t], p)$ can be abbreviated as $\mathsf{H}\,[t]\,p$, and similarly for D and X.

- Propositional connectives (but not quantifiers) may be admitted in the second argument position of H. Thus $\mathsf{H}(w, p \wedge q)$ abbreviates $\mathsf{H}(w,p) \wedge \mathsf{H}(w,q)$.

- We define $[s,t]\,f := x$ as an abbreviation for
  $\mathsf{X}\,[s,t]\,f \wedge \mathsf{H}\,[t]\,f\!:\!x$,
  which is intended to say that f is affected by change during the period $[s,t]$ and its value in t is x.

- Similarly, to indicate change over an interval represented by a constant or a variable, $w \mid f := x$ abbreviates
  $\mathsf{X}(w,f) \wedge \mathsf{H}(w, f\!:\!x)$.

- We use the customary abbreviations in the context of binary relations, for example $s \leq t$ for $s < t \vee s = t$, and $s < t < u$ for $s < t \wedge t < u$.

- We use the expression $\forall\!\!\forall\,[\phi]$ as an abbreviation for the formula formed from ϕ by enclosing it with universal quantification of all its free variables.

5 Time domain

The HGRL allows several different choices of time domain. The two primary choices are *real-valued time* and *situational time*. We shall first introduce these two main alternatives very briefly, and then specify the general assumptions about the time domain. The range of possible choices of time domain is described in an onward section.

5.1 Real-valued time

In the case of real-valued time, the time domain is chosen as the real numbers, and the relation $<$ has its usual meaning. Following the work of Ribeiro (reference), the domain of periods is chosen so that the following constructs are included:

- A finite open interval between the times s and t where $s < t$,
- The singleton set $\{s\}$ where s is a time,
- The empty set.

The interval forming function $[s,t]$ forms a period according these three intervals when $s < t$, $s = t$, and $s > t$, respectively. The relation $\mathsf{H}(w,p)$ says that p holds at every time in the period w, as well as in the two endpoints in the case of a non-point interval. The relation $\mathsf{D}(w,p)$ says that p holds in every member of w, not implying anything about the endpoints, but in addition it says that the interval is maximal. In this way, doing the same action twice in immediate succession is not the same thing as doing it once during the combined interval of time, in this logic.

The exact sense of "maximal interval", for D, is as follows. [*Have I reconstructed this correctly? /Erik*] If $\mathsf{D}([s,t],p)$ then either p is false in s, or there is some interval $[s-\varepsilon,s]$ where p is false, and similarly in or to the right of t.

This means, in particular, that
$\mathsf{H}([s,t],p) \wedge \mathsf{H}([t,u],p) \rightarrow \mathsf{H}([s,u],p)$,
but the same does *not* hold for D.

5.2 Situational time

Situational time is the same as one of the proposed semantics for the situation calculus, namely, that semantics where the situations form a tree with a specific situation $\mathsf{s0}$ as the root, and where actions are used to form descendants in the situation tree. The operator $\,.\,$ is used to form situations, so that if s is a situation and a is a proposition, which is understood as an action, then $s\,.\,a$ is a composite situation obtained from s by performing the action a. For example,

$\mathsf{s0}\,.\,\textsc{load}\,.\,\textsc{fire}$

represents the situation obtained by first doing "load" and then "fire" (in immediate succession).

In this semantics, the situation domain satisfies an induction axiom, as proposed by Reiter [reference], so that if s0 has a property P, and

$\forall a \forall s[P(s) \rightarrow P(s\,.\,a)]$

then $\forall s[P(s)]$. The function $\,.\,$ is a Herbrand function, like : and $[-,-]$. The relation $<$ is defined so that $s < t$ iff t is a direct or indirect descendant of s, but of course not $s = t$. An interval $[s,t]$ is defined as $\{u \mid s < u \leq t\}$. Note tha $[s,s]$ is the empty set in this semantics.

The relation between D and $\,.\,$ is defined by

$\mathsf{D}[s,t]a \leftrightarrow t = s\,.\,a.$

5.3 Interval relations

Allen [reference] has introduced a set of relations on intervals, and algebraic operations on those relations. These relations are applied to periods in both of the semantics which have been presented.

5.4 General properties of times and periods

To be written.

5.5 Total catalogue of possible semantics for times and periods

Reference to an onward section, not covered in the present one. To be written, based on the general properties of the previous subsection. In particular, cover integer time.

6 Chronicles

6.1 Action laws

An action law is a particular syntactic construct which is intended to characterize the effects of an action. It has the general form

$\mathsf{D}([s,t],p) \Rrightarrow \overset{s,t}{\forall\!\!\!\forall}\,[\phi]$

where the indication s, t above the $\forall\!\!\!\forall$ [] symbol means to quantify over all free variables in ϕ except s and t. Alternatively it may have the form

$\mathsf{D}(w,p) \Rrightarrow \overset{w}{\forall\!\!\!\forall}\,[\phi]$.

For example, the following action law

$\mathsf{D}(w, \textsc{load}) \Rrightarrow w \mid l := \mathsf{T}$

says that if the action LOAD is performed over a period w, then the feature l receives the value T over that period.

Specific restrictions on the form of action laws are introduced as prerequisites for the definitions of intended models of scenario descriptions. Unrestricted action laws are difficult to deal with.

6.2 Chronicles

A chronicle is a structure consisting of action laws and logic formulae in HGRL. Different varieties of chronicles will be defined successively; various deduction methods and various definitions of intended models make different assumptions about the exact forms of chronicles that they operate on.

In the very simplest case, a chronicle is simply a pair $\langle \textsc{law}, \textsc{scen} \rangle$, where LAW is a set of action laws, and SCEN is a set of HGRL formulae without restrictions which describe the scenario in question. The ϕ component in each action law is then an HGRL formula, without restrictions. A number of deduction methods are defined in such a way that they can be applied to chronicles of this form, but at present we are not able to say much about their correctness properties.

In order to obtain some concrete results, certain restrictions have to be imposed on the syntactic forms of action laws and scenario descriptions. One case of chronicles has been analyzed with considerable detail, namely so-called $\mathcal{K}$-**IA** chronicles. This class corresponds to the following assumptions:

- Complete knowledge about what actions have been performed.
- Complete knowledge about the set of possible outcomes of each kind of action.
- All statements correct (no unreliable observations etc).
- All actions are performed in sequence.
- Nondeterministic actions are allowed.
- Actions may have effects during an extended period of time, but that period must be finite.
- All effects of actions are specified explicitly in the action laws. The world is strictly inert, there are no indirect effects, and no ramification.
- The effects of an action may depend on the starting state, that is, the state of the world at the time when the action starts. However, the actions effects (or set of possible effects) must be known for all possible starting states.

The requirements for $\mathcal{K}$-**IA** are primarily that action laws must be designed in a certain way. In the situation-calculus case this is sufficient. If metric time is used, then additional structure and constraints must be imposed on the scenario description. For the full details about this topic, see section [to be written].

Author affiliations

Erik Sandewall: Department of Computer and Information Science, Linköping University, Sweden

Antonio Porto: Department of Computer Science, New University of Lisbon, Portugal