Symbolic Trajectories

Pattern Matching - Requirements and First Results

Fabio Valdés

21. März 2012

Contents

The operator `matches`

- Checks whether a pattern matches a moving label

The operator `matches`

- Checks whether a pattern matches a moving label
- Syntax:

The operator `matches`

- Checks whether a pattern matches a moving label
- Syntax:
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{pattern}} \rightarrow \underline{\texttt{bool}}$

The operator `matches`

- Checks whether a pattern matches a moving label
- Syntax:
  - <u>`mlabel`</u> × <u>`pattern`</u> → <u>`bool`</u>
  - <u>`mlabel`</u> × <u>`text`</u> → <u>`bool`</u>

Moving Label

Originally a stream of tuples of the form

```
( (instant start, instant end,
bool left_closed, bool right_closed)
ulabel unit_label)
```

Moving Label

Originally a stream of tuples of the form

```
( (instant start, instant end,
bool left_closed, bool right_closed)
ulabel unit_label)
```

Simplification

```
(periods period, ulabel unit_label)
```

Pattern

a simple example

```
(thursday _) (_ at_work) * (_ at_home)
(2011-04-02#19:09:00 _)
```

Pattern

a simple example

```
(thursday _) (_ at_work) * (_ at_home)
(2011-04-02#19:09:00 _)
```

variables may be associated to unit/sequence patterns

```
X (_ at_home) Y * Z (monday _)
```

Pattern

a simple example

```
(thursday _) (_ at_work) * (_ at_home)
(2011-04-02#19:09:00 _)
```

variables may be associated to unit/sequence patterns

```
X (_ at_home) Y * Z (monday _)
```

conditions can be added for variables

```
X (_ at_home) Y * Z (monday _) //
X.start > 2011-10-23, Y.card > 25,
Z.label = at_university
```

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways
  - year, month, day, hour, minute, second

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways
  - year, month, day, hour, minute, second
    - e.g., 2011-04-02#19:09 or 2012

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways
  - year, month, day, hour, minute, second
    - e.g., 2011-04-02#19:09 or 2012
  - ranges or halfopen ranges of date or time

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways
  - year, month, day, hour, minute, second
    - e.g., 2011-04-02#19:09 or 2012
  - ranges or halfopen ranges of date or time
    - e.g., 2011-04-30/2011-05-15#22 or 2011-04/

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways
  - year, month, day, hour, minute, second
    - e.g., 2011-04-02#19:09 or 2012
  - ranges or halfopen ranges of date or time
    - e.g., 2011-04-30/2011-05-15#22 or 2011-04/
  - semantic time ranges

Unit Patterns

- `()`, an abbreviation for `(_ _)`, matches any unit
- the time interval can be entered in different ways
  - year, month, day, hour, minute, second
    - e.g., 2011-04-02#19:09 or 2012
  - ranges or halfopen ranges of date or time
    - e.g., 2011-04-30/2011-05-15#22 or 2011-04/
  - semantic time ranges
    - e.g., thursday, march, morning

Sequence Patterns

- $\star$ matches any sequence

Sequence Patterns

- $*$ matches any sequence
- $+$ matches any sequence with at least one unit

Sequence Patterns

- $\star$ matches any sequence
- $+$ matches any sequence with at least one unit
- `((time label))` matches a continuous sequence of units that fulfill the condition(s)

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then

  are also patterns

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then
  - $[\, p_1 \mid p_2 \,]$

are also patterns

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then
  - $[\,p_1 \mid p_2\,]$
  - $[\,p_1\,]+$

  are also patterns

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then
  - $[\, p_1 \mid p_2 \,]$
  - $[\, p_1 \,] +$
  - $[\, p_1 \,] \ast$

  are also patterns

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then
  - $[\, p_1 \mid p_2 \,]$
  - $[\, p_1 \,] +$
  - $[\, p_1 \,] \star$

  are also patterns
- Label hierarchies

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then
  - $[\, p_1 \mid p_2 \,]$
  - $[\, p_1 \,] +$
  - $[\, p_1 \,] \star$

  are also patterns
- Label hierarchies
  - e.g., `leisure` $\rightarrow$ `building` $\rightarrow$ `museum` $\rightarrow$ `...`

Features to be Added

- Regular expressions If p_1 and p_2 are patterns, then
  - $[\,p_1 \mid p_2\,]$
  - $[\,p_1\,]+$
  - $[\,p_1\,]\star$

  are also patterns
- Label hierarchies
  - e.g., `leisure` $\rightarrow$ `building` $\rightarrow$ `museum` $\rightarrow$ `...`
  - then, the pattern `(_ museum)` would match the moving label `(_ leisure)`

The Operator `apply`

- Syntax:

The Operator `apply`

- Syntax:
  - <u>`mlabel`</u> $\times$ <u>`rule`</u> $\rightarrow$ <u>`stream`</u>`(`<u>`mlabel`</u>`)`

The Operator `apply`

- Syntax:
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{rule}} \rightarrow \underline{\texttt{stream}}(\underline{\texttt{mlabel}})$
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{text}} \rightarrow \underline{\texttt{stream}}(\underline{\texttt{mlabel}})$

The Operator `apply`

- Syntax:
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{rule}} \to \underline{\texttt{stream}}(\underline{\texttt{mlabel}})$
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{text}} \to \underline{\texttt{stream}}(\underline{\texttt{mlabel}})$
- For each way the pattern matches, one rewritten version of it is returned

The Operator `apply`

- Syntax:
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{rule}} \rightarrow \underline{\texttt{stream}}(\underline{\texttt{mlabel}})$
  - $\underline{\texttt{mlabel}} \times \underline{\texttt{text}} \rightarrow \underline{\texttt{stream}}(\underline{\texttt{mlabel}})$
- For each way the pattern matches, one rewritten version of it is returned
- If there is no match, the stream is empty