Advanced Software Engineering
Interactive Exercise 1: **Constraint DSLs**

Christoph Czepa

Software Architecture Research Group

Faculty of Computer Science

University of Vienna

http://cs.univie.ac.at/swa

Example: **CoLa**

```
Constraint loan_approval01 {
  loan_approval_clerk.finished precedes loan_approval_supervisor.started
  Documentation: "Clerks must first approve a loan application, only then
are supervisors allowed to further handle the application (reason: cost
reduction)."
  Mandatory: yes
}

Constraint loan_approval02 {
  after loan_amount > 100000 [ loan_approval.finished leads to
loan_approval_supervisor.started ]
  Documentation: "If the loan amount exceeds 100.000 EUR, then also a
supervisor must approve."
  Mandatory: yes
}
```

Example: LTL-SNL

```
Constraint loan_approval01 {
  not loan_approval_supervisor.started until loan_approval_clerk.finished
  Documentation: "Clerks must first approve a loan application, only then
are supervisors allowed to further handle the application (reason: cost
reduction)."
  Mandatory: yes
}

Constraint loan_approval02 {
  globally(loan_amount > 100000 implies globally(loan_approval.finished
implies finally loan_approval_supervisor.started)
  Documentation: "If the loan amount exceeds 100.000 EUR, then also a
supervisor must approve."
  Mandatory: yes
}
```

Example: TQL

```
Constraint loan_approval01 {
  initial truth value: temporarily satisfied
  permanently satisfied: loan_approval_clerk.finished
  permanently violated: not loan_approval_supervisor.started until
loan_approval_clerk.finished
  Documentation: "[...]"
  Mandatory: yes
}

Constraint loan_approval02 {
  initial truth value: temporarily satisfied
  temporarily satisfied: loan_amount > 100000 leads-to every
loan_approval.finished leads to loan_approval_supervisor.started
  temporarily violated: loan_amount > 100000 leads-to every
loan_approval.finished
  Documentation: "[...]"
  Mandatory: yes
}
```

CoLa
Operators

- A leads-to B: *Whenever A happens, B must happen eventually*
  - [A] – temporarily violated
  - [A, B] – temporarily satisfied
  - [A, B, A] – temporarily violated
  - [A, B, A, C] – temporarily violated
  - [A, B, A, C, B] – temporarily satisfied
- A precedes B: *B is only allowed to happen if A already has happened*
  - [C] – temporarily satisfied
  - [C, A] – permanently satisfied
  - [C, B] – permanently violated
- A occurs
  - [C] – temporarily violated
  - [C, A] – permanently satisfied
- A never occurs
  - [C] – temporarily satisfied
  - [C, A] – permanently violated

CoLa
Operators and Scopes

- After C [A leads to B]
  - [A] – temporarily satisfied
  - [A, C] – temporarily satisfied
  - [A, C, B] – temporarily satisfied
  - [A, C, B, A] – temporarily violated
  - [A, C, B, A, B] – temporarily satisfied
  - [A, C, B, A, B, A] – temporarily violated
- Between C and D [A leads to B]
  - [A] – temporarily satisfied
  - [A, C, E, A] – temporarily satisfied
  - [A, C, E, A, D] – permanently violated
- After C until D [A leads to B]
  - [A] – temporarily satisfied
  - [A, C, E, A] – temporarily violated
  - [A, C, E, A, D] – permanently violated

LTL-SNL
Operators

- globally p
  - *p must always hold*
- finally p
  - *p must eventually hold*
- next p
  - *p must hold at the next state*
- p_1 until p_2
  - *p_1 must hold until p_2 holds*
- p_1 weak-until p_2
  - *p_1 must hold until p_2 holds or globally not p_1*
- Boolean Logic Operators (not, and, or, implies)

LTL-SNL
Operator Precedence

- finally, globally, next, not
- until
- weak-until
- and
- or
- implies

$\uparrow$ Strongest Binding

$\downarrow$ Weakest Binding

finally A implies not B or P until C   (finally A) implies (((not B) or P) until C)

LTL-SNL
Example 1

globally(A implies finally B)

Every time A happens, B must follow somewhere thereafter

- temporarily satisfied
  - [A, B]
  - [C]
  - [C, B]
  - [C, B, A, C, B]
- temporarily violated
  - [A, B, A]
  - [A, C]
  - [C, B, A]
  - [C, B, A, C, B, A]

LTL-SNL
Example 2

$$\underbrace{\text{finally T implies}}_{\text{If T eventually happens, then}}\ \underbrace{\text{not S until (J or T)}}_{\text{S is not allowed to happen until J or T occurs}}$$

As long as T has not occurred: temporarily satisfied

As soon as T occurs for the first time:

- trace fulfills *not S until (J or T)* → permanently satisfied
  - [A, T]
  - [A, J, S, T]
- otherwise → permanently violated
  - [A, S, T]
  - [A, S, J, T]

TQL
Operators

- every *e*
  - *fire for every occurrence of e*
- not *e*
  - *start as true and change to false once e fires*
- not e_1 until e_2
  - *becomes true if* e_1 *does not occur until* e_2 *occurs*
- e_1 and e_2
  - *becomes true when both become true*
- e_1 or e_2
  - *becomes true when at least one becomes true*
- e_1 leads-to e_2
  - *becomes true when* e_2 *is finally followed by* e_2

TQL

Operator Precedence

- every, not
- until
- and
- or
- leads-to

Strongest Binding

Weakest Binding

A leads-to not C until B   A leads-to ((not C) until B)

TQL

Example 1

initial truth value: temporarily satisfied

temporarily satisfied query: every(A leads-to B)

temporarily violated query: every A

- temporarily satisfied
  - [A, B]
  - [C]
  - [C, B]
  - [C, B, A, C, B]
- temporarily violated
  - [A, B, A]
  - [A, C]
  - [C, B, A]
  - [C, B, A, C, B, A]

TQL
Example 2

initial truth value: temporarily satisfied

permanently violated query: not M and not Q until K leads-to M

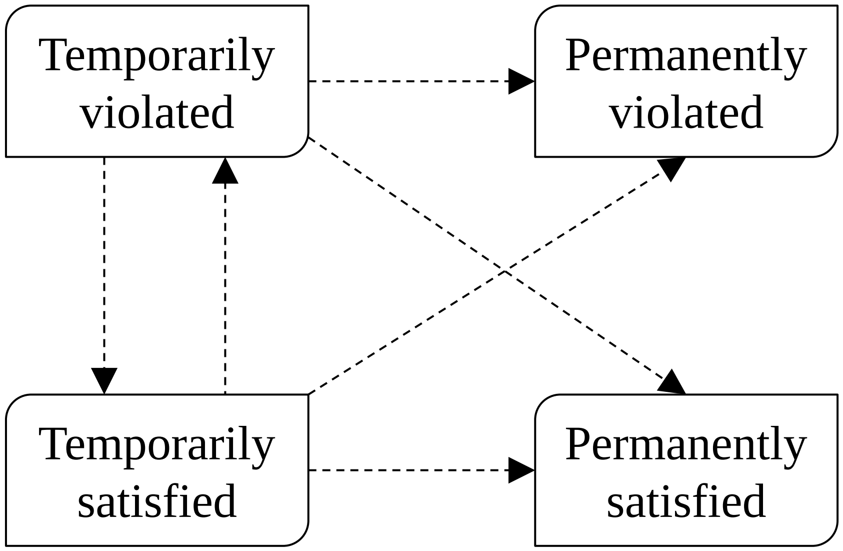
permanently satisfied query: not K until Q

- permanently violated
  - [A, K, A, M]
  - [A, K, A, Q, M]
- permanently satisfied
  - [A, Q]
- temporarily satisfied
  - [A]
  - [A, M, K, M]

TQL
Examples for operator *every*

- every(A leads-to B)
  - [A, B] → fires
  - [A, B, B]
  - [A, B, B, A]
  - [A, B, B, A, B] → fires
- A leads-to every B
  - [A, B] → fires
  - [A, B, B] → fires
- every(A leads-to not C and every(B))
  - [A, B] → fires
  - [A, B, A, C, B]
  - [A, B, A, C, B, A, B] → fires
  - [A, B, A, C, B, A, B, B] → fires

Possible Truth Value Changes


Temporarily violated
Permanently violated
Temporarily satisfied
Permanently satisfied