A Temporal Logic for Programs

Steffen Schlager

3rd KeY Workshop
Königswinter, June 2004

Motivation

Dynamic Logic (DL)

- **“talks” about final state of program**
- **not useful for non-terminating programs**
- **does not allow reasoning about temporal properties**
- **“waste”: symbolic execution computes all intermediate program states (trace) but throws away everything except for the final state!**

Background

First approach [Beckert & Schlager, 2001]

- **Extension of DL with additional modalities “preserves”, “throughout”, and “at least once”**
- **Example:** $x > 0 \rightarrow [[\texttt{while (true)}\, x\text{++}]]x > 0$
- **Calculus for JavaCard-DL in [Beckert & Mostowski, 2003] implemented in KeY**

Deficiencies

- **Each modality strictly bound to one program**
- **Modalities cannot be combined as usual in temporal logics**

  Example: $\Box(x < 0 \longrightarrow \Diamond x > 0)$

  "It must hold in all states that if x becomes negative eventually it will become positive"
- **Expressing above property requires new modality**

Idea

- **Combine ideas from Dynamic Logic and Temporal Logic**
- **Decouple modal operators and programs**
- **Program defines structure which temporal formula is evaluated in**
- **Example:**

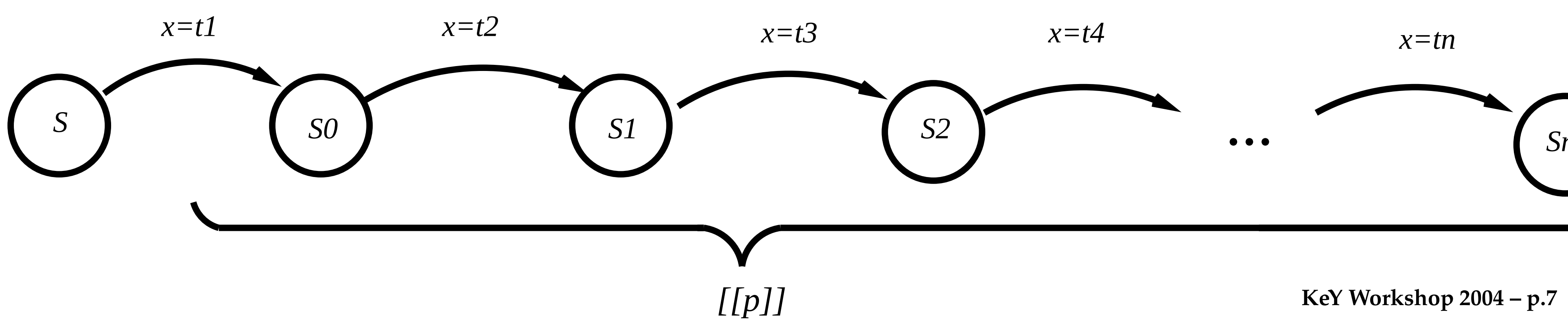
  $\forall x.(i \doteq x \rightarrow [\![\texttt{ while (true) i++}]\!] \Box (x < 0 \rightarrow \Diamond x > 0))$
- **Semantics of $[\![p]\!]$ is the (in-)finite trace of program p**

Syntax of Dynamic Temporal Logic (DTL)

- **if** $\phi \in F(FOL)$ **then** $\phi \in F(DTL)$
- **if** $\phi, \psi \in F(DTL)$**,** p **is a program, and** x **is a variable then**
  - $\Box\phi, \Diamond\phi, \phi\mathbf{U}\psi \in F(DTL)$
  - $\neg\phi, \phi \wedge \psi \in F(DTL)$
  - $[\![p]\!]\phi \in F(DTL)$ **if** ϕ **contains an unbound modal operator**
  - $\forall x.\phi \in F(DTL)$

Semantics of DTL

- $I_s([\![p]\!]) = (s_0, s_1, \ldots, s_n)$ **where** s **is initial state**
- $I_s(\mathtt{x{=}t}) = (s^t_x)$ **(transitions only by assignments)**
- $I_s(\mathtt{u;\ v}) = I_s(\mathtt{u}) \circ I_{last(I_s(\mathtt{u}))}(\mathtt{v})$
- **for** $\phi \in F(FOL)$**:** $s \models [\![p]\!]\phi$ **iff** $s \models \phi$
- $s \models [\![p]\!]\phi \mathrm{U} \psi$ **iff for a** s_i **with** $0 \leq i \leq n$ **holds** $s_i \models (s_{i+1}, s_{i+2}, \ldots, s_n)$ **and for all** s_j **with** $0 \leq j < i$ **holds** $s_j \models (s_{j+1}, s_{j+2}, \ldots, s_{i-1})$
- $s \models [\![p]\!]\Box\phi$ **iff for all** s_i **with** $0 \leq i \leq n$ **holds** $s_i \models (s_{i+1}, s_{i+2}, \ldots, s_n)$
- $s \models [\![p]\!]\Diamond\phi$ **iff for a** s_i **with** $0 \leq i \leq n$ **holds** $s_i \models (s_{i+1}, s_{i+2}, \ldots, s_n)$



Examples

- $\Box \mathit{false}$ **holds only in final states**
- **DL modalities can be expressed**
  - $[p]\phi \equiv [\![p]\!]\Box(\Box \mathit{false} \rightarrow \phi)$
  - $\langle p\rangle\phi \equiv [\![p]\!]\Diamond(\Box \mathit{false} \land \phi)$
- $[\![\texttt{i=1; while (true) i++}]\!]\forall x.\Box(i \doteq x \rightarrow \Diamond i \doteq 2x)$

A Sequent Calculus for DTL

Assignment Rule for "throughout"

$$\frac{\Gamma \vdash \phi,\ \Delta \qquad \Gamma \vdash \{x := t\}[[\omega]]\phi,\ \Delta}{\Gamma \vdash [[x = t;\omega]]\phi,\ \Delta}$$

Assignment Rule for $\Box$

$$\frac{\Gamma \vdash \{x := t\}[\![\omega]\!]\phi,\ \Delta \qquad \Gamma \vdash \{x := t\}[\![\omega]\!]\Box\phi,\ \Delta}{\Gamma \vdash [\![x = t;\omega]\!]\Box\phi,\ \Delta}$$

A Sequent Calculus for DTL

Concatenation rule for "at least once

$$\frac{\Gamma \vdash \langle\!\langle \alpha \rangle\!\rangle \phi,\ \langle \alpha \rangle \langle\!\langle \beta \rangle\!\rangle \phi,\ \Delta}{\Gamma \vdash \langle\!\langle \alpha; \beta \rangle\!\rangle \phi,\ \Delta}$$

General concatenation rule for DTL not possible!

Rule for special case $\phi \in F(FOL)$

$$\frac{\Gamma \vdash [\![\alpha]\!] \Diamond \phi,\ \langle \alpha \rangle [\![\beta]\!] \Diamond \phi,\ \Delta}{\Gamma \vdash [\![\alpha; \beta]\!] \Diamond \phi,\ \Delta}$$

Improving the previous concatenation rule

$$\frac{\Gamma \vdash [\![\alpha]\!]\Diamond\phi,\ \langle\alpha\rangle[\![\beta]\!]\Diamond\phi,\ \Delta}{\Gamma \vdash [\![\alpha;\beta]\!]\Diamond\phi,\ \Delta}$$

Rule requires duplicate computation of trace of α!

Similar to the rule for “at least once”

Improving the previous concatenation rule

$$\frac{\Gamma \vdash [\![\alpha]\!]\Diamond\phi,\ \langle\alpha\rangle[\![\beta]\!]\Diamond\phi,\ \Delta}{\Gamma \vdash [\![\alpha;\beta]\!]\Diamond\phi,\ \Delta}$$

Rule requires duplicate computation of trace of α!

Similar to the rule for "at least once"

Improved rule

$$\frac{\Gamma \vdash [\![\alpha]\!]\Diamond(\phi \vee (\Box \mathit{false} \wedge [\![\beta]\!]\Diamond\phi)),\ \Delta}{\Gamma \vdash [\![\alpha;\beta]\!]\Diamond\phi,\ \Delta}$$

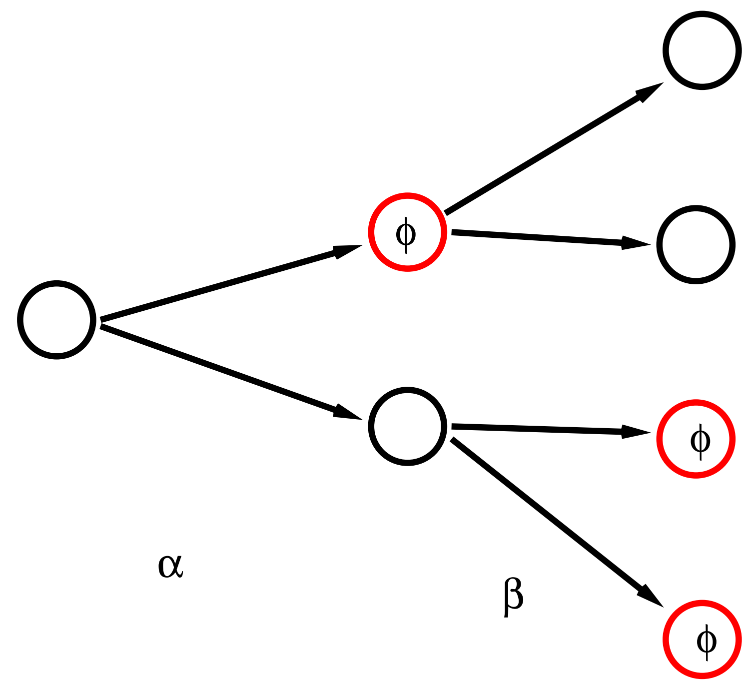
Towards a CTL-Version

Now we consider non-deterministic languages!

Semantics of $\Diamond$?

there is a path such that $\Diamond\phi$ or for all paths $\Diamond\phi$

$$\frac{\Gamma \vdash [[\alpha]]\Diamond\phi,\ \langle\alpha\rangle[[\beta]]\Diamond\phi,\ \Delta}{\Gamma \vdash [[\alpha;\beta]]\Diamond\phi,\ \Delta}$$



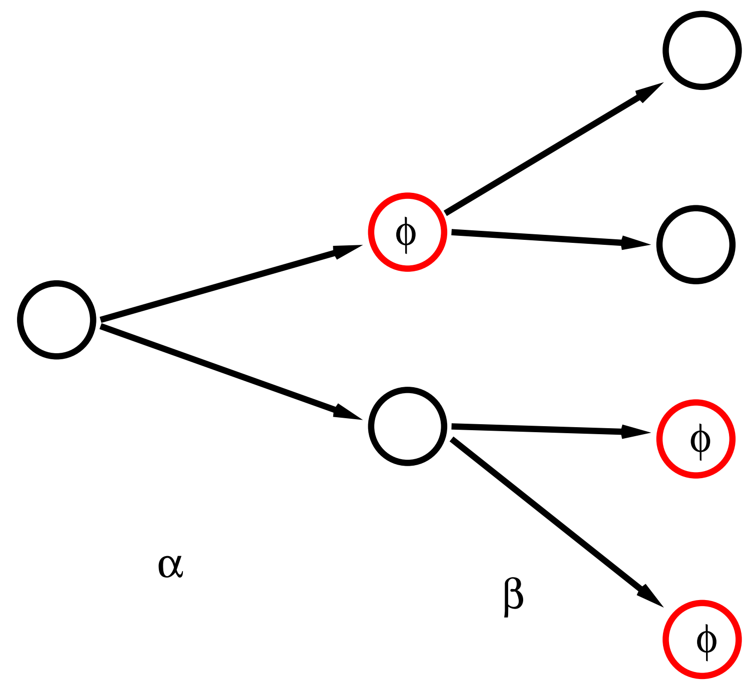
Towards a CTL-Version

Now we consider non-deterministic languages!

Semantics of $\Diamond$?

there is a path such that $\Diamond\phi$ or for all paths $\Diamond\phi$

$$\frac{\Gamma \vdash [[\alpha]]\Diamond\phi,\ \langle\alpha\rangle true \wedge [\alpha][[\beta]]\Diamond\phi,\ \Delta}{\Gamma \vdash [[\alpha;\beta]]\Diamond\phi,\ \Delta}$$



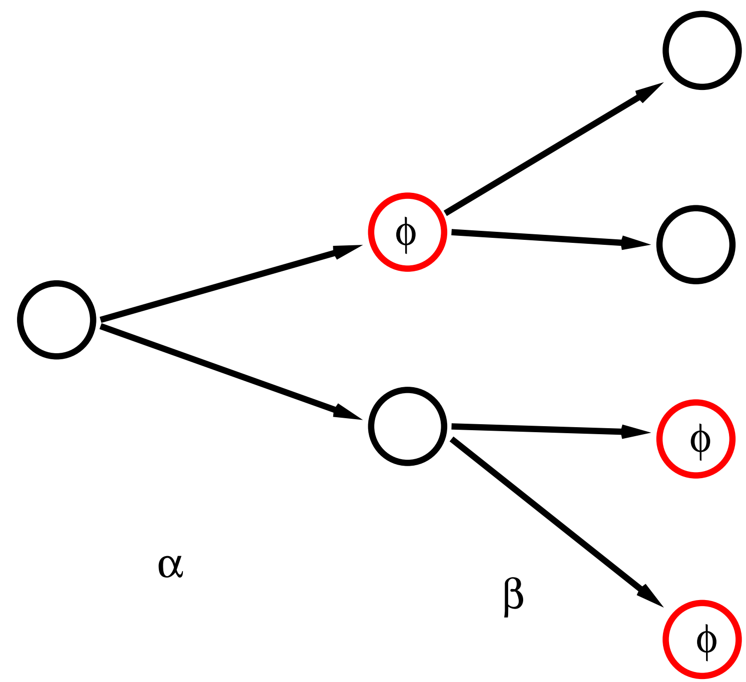
Towards a CTL-Version

Now we consider non-deterministic languages!

Semantics of $\Diamond$?

there is a path such that $\Diamond\phi$ or for all paths $\Diamond\phi$

$$\frac{\Gamma \vdash [[\alpha]]\Diamond(\phi \vee (\Box \mathit{false} \wedge [[\beta]]\Diamond\phi)),\ \Delta}{\Gamma \vdash [[\alpha;\beta]]\Diamond\phi,\ \Delta}$$



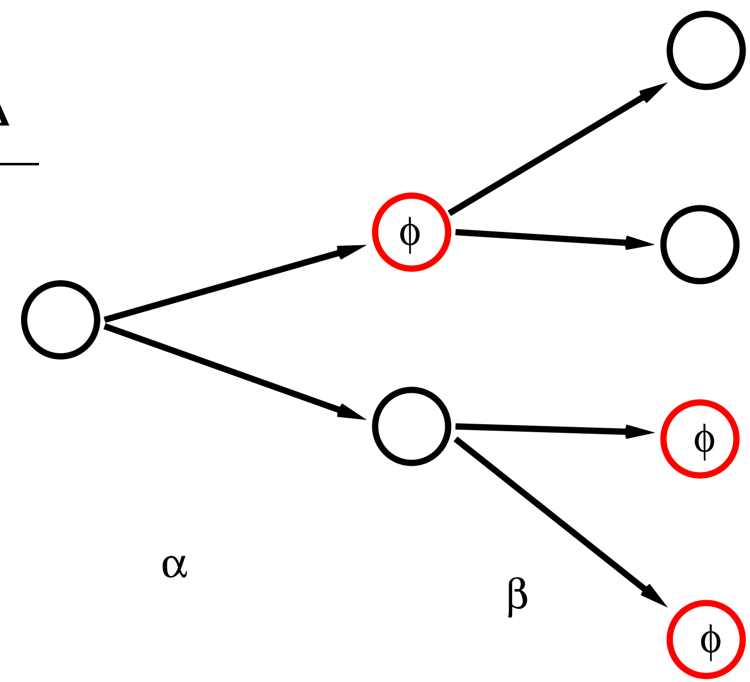
Towards a CTL-Version

Now we consider non-deterministic languages!

Semantics of $\Diamond$?

there is a path such that $\Diamond\phi$ or for all paths $\Diamond\phi$

$$\frac{\Gamma \vdash [[\alpha]]\mathbb{Q}(\Diamond(\phi \vee (\Box \mathit{false} \wedge [[\beta]]\mathbb{Q}\Diamond\phi))),\ \Delta}{\Gamma \vdash [[\alpha;\beta]]\mathbb{Q}\Diamond\phi,\ \Delta}$$



Rules for Loops

- **Similar to rules for μ-calculus**
- **Idea: identify repeats in the proof**

Example:

$$
\dfrac{i \doteq 1 \vdash i > 0 \qquad \dfrac{\dfrac{A \qquad \dfrac{\dfrac{\color{red}{i \doteq c,\ c > 0 \vdash [\![p]\!]\Box(i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)}}{i \doteq i' + 1,\ i - 1 \doteq c,\ c + 1 > 0 \vdash [\![p]\!]\Box(i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)}\ \mathit{Subst}.\{c \leftarrow c+1\}}{i \doteq i' + 1,\ i' \doteq c,\ c > 0 \vdash [\![p]\!]\Box(i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)}\ \mathit{Cut\&Weakening}}{i \doteq c,\ c > 0 \vdash [\![i\texttt{++};\ p]\!]\Box(i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)}\ \mathit{Assignm.}}{\color{red}{i \doteq c,\ c > 0 \vdash [\![p]\!]\Box(i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)}}}{\dfrac{i \doteq 1 \vdash [\![p]\!]\Box(i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)}{i \doteq 1 \vdash [\![\texttt{while (true)}\ i\texttt{++}]\!]\forall x.\Box(i \doteq x \rightarrow \Diamond i \doteq 2x)}}\ \mathit{Gen.}
$$

with

$$
A \;:=\; i \doteq i' + 1,\ i' \doteq c,\ c > 0 \vdash [\![p]\!](i \doteq x_0 \rightarrow \Diamond i \doteq 2x_0)
$$

Future Work

- Finishing work on rules for loops
- DTL for $\mathrm{PROMELA}^{+}$
  - non-deterministic constructs
  - communication via channels
  - processes and dynamic process creation
- Translating statecharts into $\mathrm{PROMELA}^{+}$ for verification of temporal properties