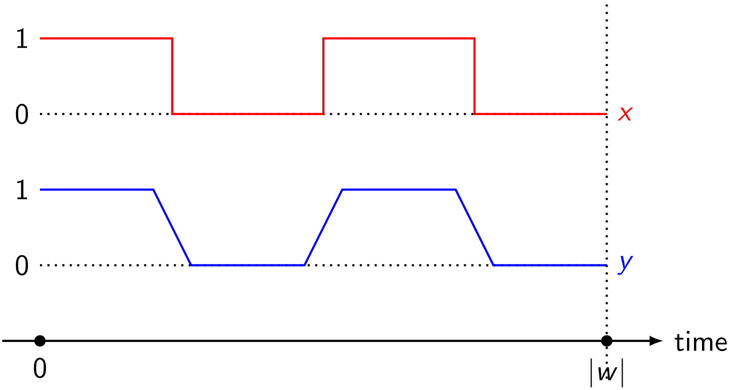
On the Quantitative Semantics of Regular Expressions over Real-Valued Signals

Alexey Bakhirkin Thomas Ferrère Oded Maler
Dogan Ulus

Verimag

IST AUSTRIA
Institute of Science and Technology

We Want to Ask Questions about Signals



- Real-valued.
- Piecewise-linear or piecewise-constant.
- Time is bounded.
- Question language – Signal Regular Expressions.

Signal Regular Expressions and Boolean Matching

Pre-Existing Work

$$\varphi \to x \geq c \mid x \leq c \mid \langle\varphi\rangle_{[a,b]} \mid \varphi_1 \cdot \varphi_2 \mid \varphi^* \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2$$

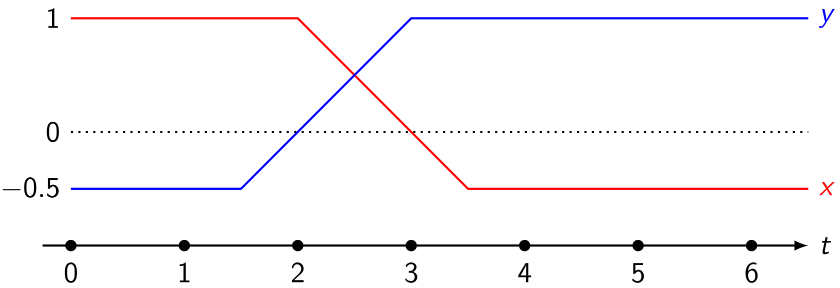
Basic question: Does the signal w match the expression φ on the interval $[t, t')$?

- $x \geq c$, if $t' > t$ and $x \geq c$ everywhere on $[t, t')$;
- $\langle\varphi\rangle_{[a,b]}$, if w matches φ on $[t, t')$, and $a \leq t' - t \leq b$;
- $\varphi \cdot \psi$, if there is t'' in $[t, t']$, s.t. w matches φ on $[t, t'')$ and ψ on $[t'', t')$.
- φ^*, if there is k (infinite disjunction can be avoided), s.t. w matches $\varphi \cdot \varphi \cdot \ldots \cdot \varphi$ (k times).
- Boolean $\wedge$ and $\vee$, as expected.

We ask: Where does the signal match the expression?
Find intervals $[t, t')$, where the signal w matches the expression φ.

Boolean Matching Example

Pre-Existing Work



Boolean semantics: Does w on $[0, 6)$ match $x \geq 0 \cdot y \geq 0$? Yes.

We ask: Where does w match $x \geq 0 \cdot y \geq 0$?
On $[t, t')$, where $t < 3$ and $t' > 2$.

If needed, can project the final answer on t.

Quantitative Matching

This Work

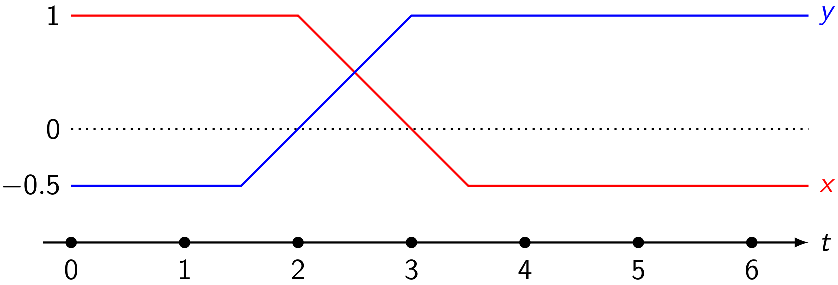
Basic question: How well does the signal w match (or does not much) the expression φ on the interval $[t, t')$?

Robustness of an expression on $[t, t')$:

- $x \geq c$, how far is x above c,
  if $t' > t$ then minimum of $x - c$ on $[t, t')$, otherwise $-\infty$;
- $\langle\varphi\rangle_{[a,b]}$, if $a \leq t' - t \leq b$ then robustness of φ on $[t, t')$, otherwise $-\infty$.
- $\varphi \cdot \psi$,
  $\max\big\{ \min\{\rho \text{ of } \varphi \text{ on } [t, t''), \rho \text{ of } \psi \text{ on } [t'', t')\} \mid t'' \in [t, t')\big\}$:
- φ^* supremum (turns into max) of robustness of φ^k for $k \geq 0$.
- For $\wedge$ and $\vee$, min and max.

Again, we compute robustness for every interval.

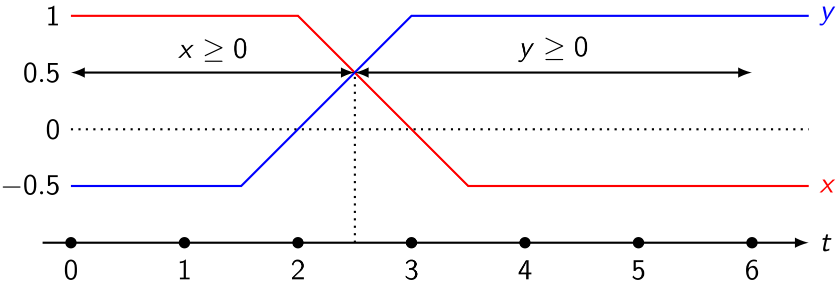
Robustness Example



- What's the robustness of $x \geq 0$ on $[0, 6)$? -0.5.
- What's the robustness of $x \geq 0 \cdot y \geq 0$ on $[0, 6)$? 0.5. We have to take $t'' = 2.5$.

Note that we compute robustness for every interval.

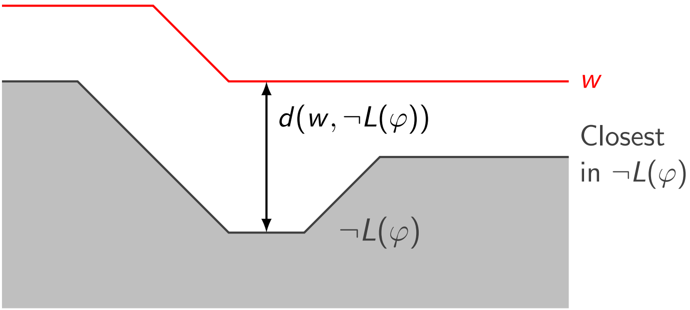
Robustness Example



- What's the robustness of $x \geq 0$ on $[0, 6)$? -0.5.
- What's the robustness of $x \geq 0 \cdot y \geq 0$ on $[0, 6)$? 0.5. We have to take $t'' = 2.5$.

Note that we compute robustness for every interval.

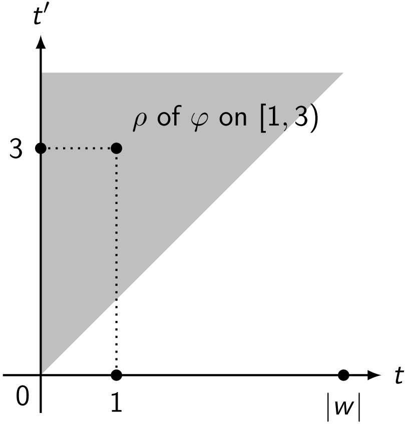
What Robustness Means



- We can shift every point of w by at most ρ, and it will still match φ.
- Assume that w matches φ on $[0, |w|)$, i.e., $\rho \geq 0$.
- Find the uniform distance to the closest signal that does not match φ.
- Robustness of φ on $[0, |w|)$ is at most this distance.

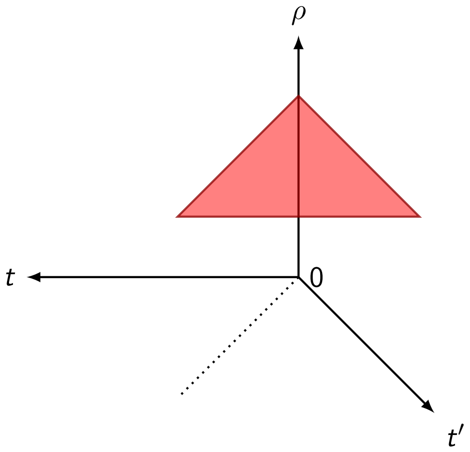
How to Compute Robustness

- Bottom-up over the expression structure.
- For every sub-expression, we want the surface:



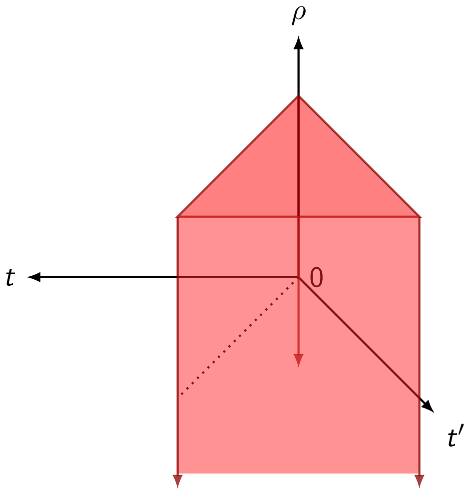
Volume Representation

- Instead of the surface, on $[t, t')$, $\rho = \cdots$.



Volume Representation

- Instead of the surface, on $[t, t')$, $\rho = \cdots$.
- Compute the "half-space" under it, on $[t, t')$, $\rho < \cdots$

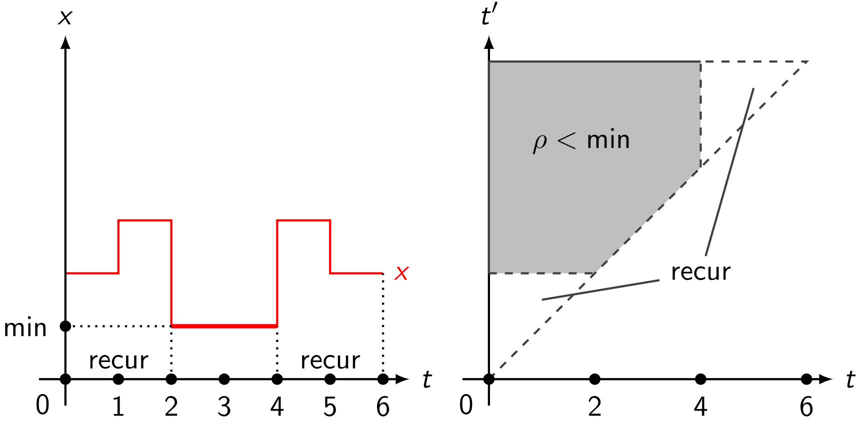


Volume Representation

- Instead of the surface, on $[t, t')$, $\rho = \cdots$.
- Compute the "half-space" under it, on $[t, t')$, $\rho < \cdots$
- Represent the robustness of φ as a set of polyhedra P_φ that fill the space under the surface.
- For $\varphi \cdot \psi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, etc., combine P_φ and P_ψ with polyhedral operations.

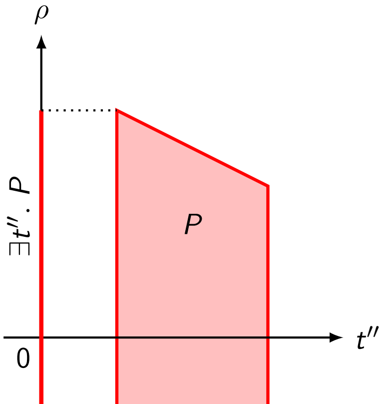
Robustness of $x \geq 0$

- Fill the volume with zones by recursively finding local minima.
- Straightforward to adapt to $x \geq c$ and $x \leq c$.
- Piecewise-linear is similar, needs convex polyhedra and a couple of extra steps.



Robustness of $\varphi \cdot \psi$

- Represent $\max\big\{ \min\{\rho(\varphi,[t,t'')), \rho(\psi,[t'',t'))\} \mid t'' \in [t,t']\big\}$.
- Given P_φ (over t, t'', ρ) and P_ψ (over t'', t', ρ).
- Insersection represents the minimum.
  $P_\varphi \wedge P_\psi$ represents $\min\{\rho(\varphi,[t,t'')), \rho(\psi,[t'',t'))\}$.
- Projection represents maximum over a dimension.
  $\exists t''.\ P_\varphi \wedge P_\psi \wedge t'' \in [t,t']$ represents $\rho(\varphi \cdot \psi, [t,t'))$.



Other Expressions

- $\langle\varphi\rangle_{[a,b]}$: $P_\varphi \wedge a \leq t' - t \leq b$.
- $\varphi \wedge \psi$: $P_\varphi \wedge P_\psi$.
- $\varphi \vee \psi$: $P_\varphi \cup P_\psi$.
- φ^*: $\varepsilon \vee \varphi \vee \varphi^2 \vee \varphi^3 \cdots$ until it stabilizes (it will).

Kleene Star is Bounded

For every signal and expression φ, exists k (depends on the signal), s.t. $\varphi \vee \varphi^2 \vee \cdots \vee \varphi^k$ is not less robust than $\varphi \vee \varphi^2 \vee \cdots \vee \varphi^k \vee \varphi^{k+1}$.

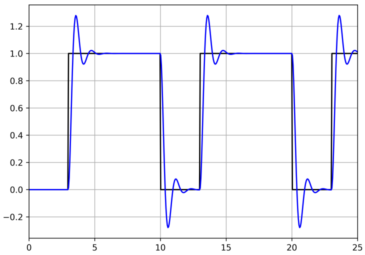
Intuitively:

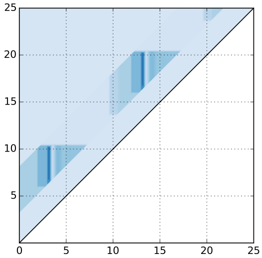
- When computing robustness of φ^k, we (look for the best way to) split the signal into k segments.
- Timing constrains in φ have resolution, they cannot distinguish between "very short" segments.
- On "very short" segments φ and φ^2 have the same robustness.
- For large enough k, some segments become short enough that splitting into $k + 1$ segments does not increase the robustness.

Experiments

Find a ringing pattern in a signal.

$$\langle x \leq 0.2 \rangle_{\leq 0.05} \cdot \langle 0.1 \leq x \leq 0.9 \rangle_{\leq 0.05} \cdot \langle 0.7 \leq x \leq 1.3 \rangle_{[0.3,\ 1]} \cdot \langle 0.9 \leq x \leq 1.1 \rangle_{[3,\ 6]}$$





With piecewise-constant interpolation:

Input length	10K	20K	40K
Execution Time (sec)	3.88	7.80	15.5
Number of output zones	156K	315K	631K

Future Work

- Optimization. Don't compute robustness for unnecessary expressions and intervals.
  Low-hanging fruit:
  - Propagate time constraints from top to bottom.
  - Discard regions with robustness below a threshold.

  Less low hanging: quantize robustness.

Future Work

- Optimization. Don't compute robustness for unnecessary expressions and intervals.
- Time robustness. Allow to violate time constraints at a cost.
- Multi-dimensional robustness. Don't add apples to oranges.

Thanks