ALLEGRO: Belief-based Programming in Stochastic Dynamical Domains
Vaishak Belle, Hector J. Levesque

OBJECTIVE
- High-level program execution, e.g., GOLOG, FLUX, are attractive and influential alternatives to planning
- especially with incomplete information, where plans with branches and loops are required
- lots of success, e.g., cognitive robotics

LIMITATIONS
- PREGO is Regression-based, appropriate for automated planning
- But no support for programs (don’t know outcomes/sensed values in advance!)
- Iterative programs = regression with an infeasible number of integrals!

ALLEGRO = ALGOL IN PREGO
- Modeler specifies basic action theory (BAT), including probabilistic beliefs and noise models
- Expressive programming language:

```plaintext
loop: if ~Empty(queue)
    then (π)p selectRequest(p);
    pickupCoffee; bringCoffee(p)
else wait!
```

where form stands for formulas:

```plaintext
form ::= (term₁ term₂) | (• form₁ form₂) | (not form)
```

where o ∈ {<, >, =}, ∈ {and, or}, and term stands for terms:

```plaintext
term ::= (exp term) | number | fluent | var (term term₂) |
       (if form term term₂)
```

To get close to the wall, for example:

```plaintext
Until (> (bel (and (< h 2) (< h 0)))) .8)
   (until (> (conf h .4) .8) (sensor2))
   (let (diff (exp h 4))
      (if (equal h 4))
      (belief (h 4))
```

QUERY MECHANISM:
- cf. paper for semantics of ALLEGRO programs, interpreted as situation-suppressed formulas.
- Successful termination after σ is expressed as:

```plaintext
D ∪ E ∪ F |= Do(δ, S₀, do(σ, S₀)).
```

EMPIRICAL EVALUATIONS
- Regression ⇒ many integrals:

- ALLEGRO scales well:

- Program termination can be studied wrt error models of sensors and actuators:

CONCLUSIONS
- A new variant of GOLOG over noise and belief
- Different from all other "probabilistic" GOLOG variants, e.g. DTGOLOG, that do not handle unobservable nondeterminism, noisy sensors, belief change, and continuous distributions
- Techniques and empirical results demonstrate promise of proposal

REFERENCES