Knowledge-Based Programs as Plans

Generating Knowledge-Based Plans

Jérôme Lang   Bruno Zanuttini   Anaëlle Wilczynski

Leiden, August 2015

Syntax

Input

- set of propositional variables $X = \{x_1, \ldots, x_n\}$ (unobservable)
- set of actions
  - ontic actions
  - epistemic actions

Knowledge-based program π:

- action, or
- sequence $\pi_1; \pi_2; \ldots; \pi_n$, or
- branching If Φ then π_1 else π_2, where Φ is a purely subjective S5 formula (Boolean combination of epistemic atoms $K\varphi$); or
- loop While Φ do π_1, where Φ is a purely subjective S5 formula.

Progression

Executing an action at a current knowledge state

Ontic action:

- progression of M^t by Σ
- $M^t = \{x_1x_2x_3, \bar{x}_1\bar{x}_2\bar{x}_3\}$
- progression by switch(x_1) :
  $M^{t+1} = \{\bar{x}_1x_2x_3, x_1\bar{x}_2\bar{x}_3\}$
- progression by reinit(x_1) :
  $M^{t+2} = \{x_1x_2x_3, \bar{x}_1x_2x_3, x_1\bar{x}_2\bar{x}_3, \bar{x}_1\bar{x}_2\bar{x}_3\}$

Epistemic action:

- progression of M^t by observation ω received
- action test($x_1 \wedge x_2$), observation $\neg(x_1 \wedge x_2)$:
  $M^{t+3} = \{\bar{x}_1x_2x_3, x_1\bar{x}_2\bar{x}_3, \bar{x}_1\bar{x}_2\bar{x}_3\}$

Regression

Determining which states could lead to a current knowledge state if we progress them by a specified action

Ontic action:

- $Reg(\mathbf{K}(v \rightarrow u), switch(u)) = \mathbf{K}(v \rightarrow \neg u)$
- $Reg(\mathbf{K}(u \wedge v), repair(u)) = \mathbf{K}v$

Epistemic action:

- states from which the progression by any possible observation ω leads to the current state
- $Reg(\mathbf{K}v \vee \mathbf{K}\neg v, test(u \wedge v))$
  $= \mathbf{K}v \vee \mathbf{K}(v \rightarrow u) \vee \mathbf{K}(\neg u \wedge v) \vee \mathbf{K}\neg v$
  $= \mathbf{K}v \vee \mathbf{K}(v \rightarrow u)$

Knowledge-based planning problems

- initial knowedge state initial M^0 :
  - possibly $K\top$
  - must contain the true initial state
- goal knowledge state G
- π valid plan if
  - terminates
  - for every possible sequence of states $s^0 \in M^0 \ldots s^{\text{final}} \in M^{\text{final}}$ we have $M^{\text{final}} \models G$

Case-of KBPs

stop $\leftarrow$ *false*
REPEAT
CASE

$\mathbf{K}\varphi_1 : \alpha_1$

$\mathbf{K}\varphi_2 : \alpha_2$

$\ldots$

$\mathbf{K}\varphi_m : \alpha_m$

END
UNTIL $\mathbf{K}$*stop*

- fully expressive: if there is a valid KBP of depth k there is also a Case-of KBP of depth k

Generate a KBP

Two characteristics for the generating algorithms:

- Direction used to find a plan
  - Progression: from the initial state to the goal
  - Regression: from the goal to the initial state
- Treatment of actions
  - Synchronously: all the actions are treated in parallel
    $\rightarrow$ Width first search
  - Asynchronously: we go further by choosing an action to treat one by one
    $\rightarrow$ Depth first search

$\rightarrow$ Goal in DKS formula: $G = \mathbf{K}\varphi_1 \vee \cdots \vee \mathbf{K}\varphi_n$
$\rightarrow$ Returned plan: Case-of KBP

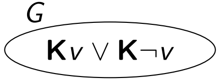
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*



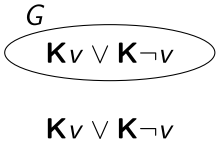
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*



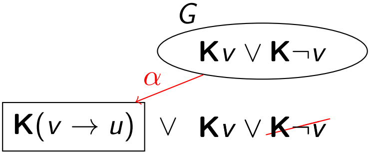
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α



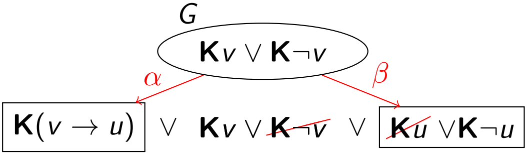
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β



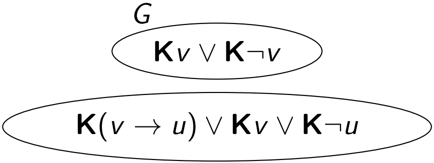
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β



Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β

G

$\mathbf{K}v \vee \mathbf{K}\neg v$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v \vee \mathbf{K}\neg u$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v \vee \mathbf{K}\neg u$

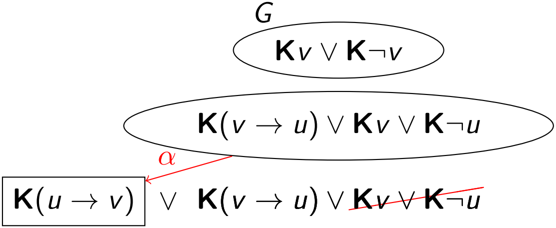
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β
- $\mathbf{K}(u \rightarrow v)$: α



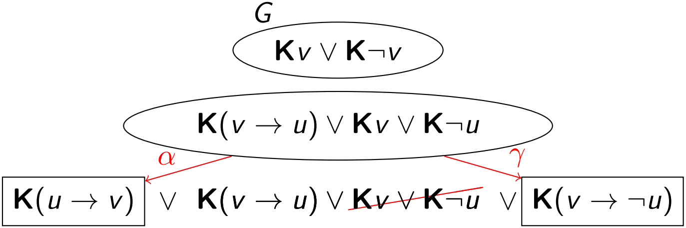
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β
- $\mathbf{K}(u \rightarrow v)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ



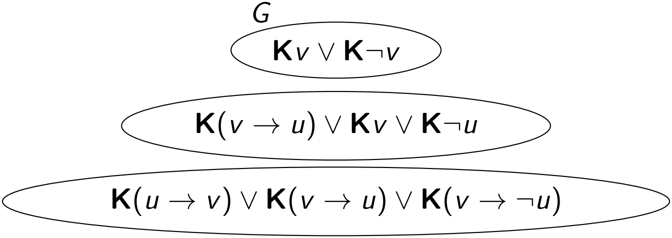
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = \mathit{test}(u \wedge v); \beta = \mathit{test}(u \leftrightarrow v); \gamma = \mathit{switch}(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β
- $\mathbf{K}(u \rightarrow v)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ



Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}\neg u$: β
- $\mathbf{K}(u \rightarrow v)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ

G

$$\mathbf{K}v \vee \mathbf{K}\neg v$$

$$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v \vee \mathbf{K}\neg u$$

$$\mathbf{K}(u \rightarrow v) \vee \mathbf{K}(v \rightarrow u) \vee \mathbf{K}(v \rightarrow \neg u)$$

$$\mathbf{K}(u \rightarrow v) \vee \mathbf{K}(v \rightarrow u) \vee \mathbf{K}(v \rightarrow \neg u)$$

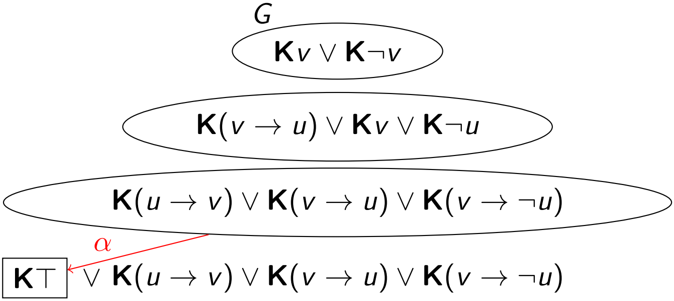
Generation by synchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u) : \alpha$
- $\mathbf{K}\neg u : \beta$
- $\mathbf{K}(u \rightarrow v) : \alpha$
- $\mathbf{K}(v \rightarrow \neg u) : \gamma$
- $\mathbf{K}\top : \alpha$



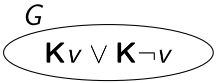
Generation by asynchronous regression

$$G = \mathbf{K}v \lor \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \land v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*



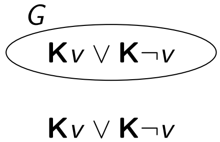
Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*



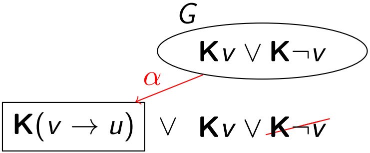
Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u) : \alpha$



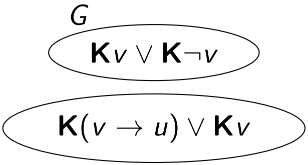
Generation by asynchronous regression

$$G = \mathbf{K}v \lor \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \land v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \to u)$: α



Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α

G

$\mathbf{K}v \vee \mathbf{K}\neg v$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v$

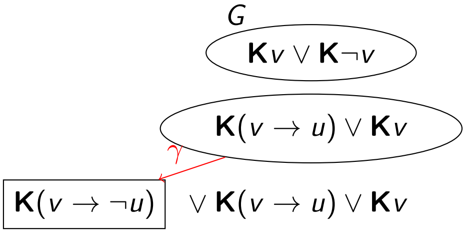
Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = \mathit{test}(u \wedge v); \beta = \mathit{test}(u \leftrightarrow v); \gamma = \mathit{switch}(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ



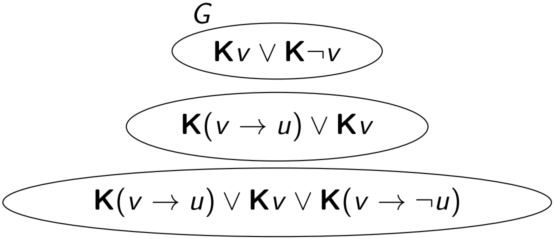
Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ



Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ

G

$\mathbf{K}v \vee \mathbf{K}\neg v$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v \vee \mathbf{K}(v \rightarrow \neg u)$

$\mathbf{K}(v \rightarrow u) \vee \mathbf{K}v \vee \mathbf{K}(v \rightarrow \neg u)$

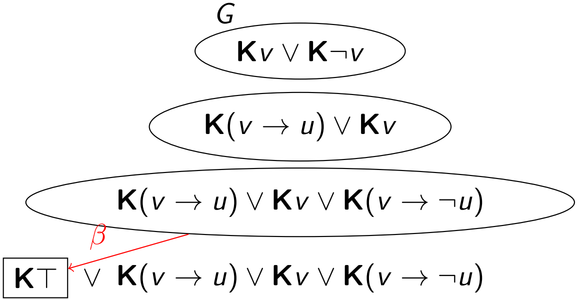
Generation by asynchronous regression

$$G = \mathbf{K}v \vee \mathbf{K}\neg v \qquad\qquad I = \top$$

$$A = \{\alpha = test(u \wedge v); \beta = test(u \leftrightarrow v); \gamma = switch(u)\}$$

Cases:

- $\mathbf{K}v$: *stop*
- $\mathbf{K}\neg v$: *stop*
- $\mathbf{K}(v \rightarrow u)$: α
- $\mathbf{K}(v \rightarrow \neg u)$: γ
- $\mathbf{K}\top$: β



Heuristic functions

Choice of the action to regress (or progress) with
$\rightarrow$ Choose the most "informative" action

- h_1: be closer to the initial state
  $\rightarrow$ optimal in the number of epistemic actions
  $M^0 = \{uv, u\bar{v}, \bar{u}v, \bar{u}\bar{v}\}$ current state $\Gamma = \mathbf{K}v \vee \mathbf{K}\neg v$
  $Reg(\Gamma, \alpha) = \mathbf{K}(v \rightarrow u) \Rightarrow h_1(\alpha) = 1$
  $Reg(\Gamma, \beta) = \mathbf{K}u \vee \mathbf{K}\neg u \Rightarrow h_1(\beta) = 2$
- h_2: be closer to the initial state and go away as much as possible from the current state
  $\rightarrow \arg\max h_2 \subseteq \arg\min h_1$
  $M^0 = \{uv, u\bar{v}, \bar{u}v, \bar{u}\bar{v}\}$ current state $\Gamma = \mathbf{K}(v \rightarrow u) \vee \mathbf{K}v$
  $Reg(\Gamma, \beta) = \mathbf{K}(u \rightarrow v) \Rightarrow h_1(\beta) = 1$ and $h_2(\beta) = 1$
  $Reg(\Gamma, \gamma) = \mathbf{K}(v \rightarrow \neg u) \Rightarrow h_1(\gamma) = 1$ and $h_2(\gamma) = 2$

Results

- Until now, only small instances can be handled
- Progression clearly faster than regression
- More succintness in Case-of KBPs generated by progression
- Depth first search: faster even if the synchronous progression/regression algorithm is optimal in the depth of the plan

Conclusion

- Regression: useful in order to characterize all the possible previous states but costly...
- Progression: faster but classical in planning
- still far from the most known classical planners (Contingent-FF, HSP)...
- ... but permits a more expressive formulation

Future work

- Properties of Case-of KBPs
- Improve the efficiency of generating KBPs by regression: find more efficient heuristics, improve implementation
- More comparisons with classical planning algorithms
- Probabilistic and multi-agent cases