$Breezy(\vec{s}) \Rightarrow Pit(\vec{s}+\hat{x}) \vee Pit(\vec{s}-\hat{x}) \vee Pit(\vec{s}+\hat{y}) \vee Pit(\vec{s}-\hat{y})$

$\neg Breezy(\vec{s}) \Rightarrow \neg Pit(\vec{s}+\hat{x}) \wedge \neg Pit(\vec{s}-\hat{x}) \wedge \neg Pit(\vec{s}+\hat{y}) \wedge \neg Pit(\vec{s}-\hat{y})$

$Safe(\vec{s}) \Rightarrow \neg Pit(\vec{s})$

$Breezy(\vec{a})$

$At(agent, \vec{s}) \Rightarrow$

Knowledge base:

- Pervasive rules: (CNF)

  Store in 3D array
  1st dim: Clauses
  2nd dim: literals
  3rd dim: Negation, type, value.

  E.g. $(P(x) \vee x) \wedge (\neg x \vee a)$

  $\Rightarrow \{P(x) \vee x, \neg x \vee a\}$

  $\Rightarrow \begin{pmatrix} P(x), & x \\ \neg x, & a \end{pmatrix}$

  $\Rightarrow \begin{pmatrix} \begin{pmatrix} 0 \\ func \\ *P(x) \end{pmatrix}, & \begin{pmatrix} 0 \\ var \\ *x \end{pmatrix} \\ \begin{pmatrix} 1 \\ var \\ *x \end{pmatrix} & \begin{pmatrix} 0 \\ const \\ a \end{pmatrix} \end{pmatrix}$

- World rules

  Store in 3D array of booleans.
  1st, 2nd dim = x, y coordinate
  3rd dim (bitwise) predicate - 64 predicates to work with

Then the knowledge can be in
  1st dim: clause
  2nd dim: literals.
  3rd dim: x, y, z coordinate of storage area.

Convert above rules to CNF

$[\neg Breezy(\vec{s}) \vee Pit(\vec{s}+\hat{x}) \vee Pit(\vec{s}-\hat{x}) \vee Pit(\vec{s}+\hat{y}) \vee Pit(\vec{s}-\hat{y})]$

$[Breezy(\vec{s}) \vee \neg Pit(\vec{s}-\hat{x})]$

$[Breezy(\vec{s}) \vee \neg Pit(\vec{s}+\hat{x})]$

$[Breezy(\vec{s}) \vee \neg Pit(\vec{s}-\hat{y})]$

$[Breezy(\vec{s}) \vee \neg Pit(\vec{s}+\hat{y})]$

$[\neg Safe(\vec{s}) \vee \neg Pit(\vec{s})]$ $Safe(\vec{s}) \Rightarrow \neg Pit(\vec{s})$

Static Rules

$[Breezy(\vec{s}) \vee \neg Pit(\vec{s}+\hat{y})]$

$[\neg Safe(\vec{s}) \vee \neg Pit(\vec{s})]$ $\quad Safe(\vec{s}) \Rightarrow \neg Pit(\vec{s})$

$[Safe(\vec{s}) \vee Pit(\vec{s})]$ $\quad \neg Safe(\vec{s}) \Rightarrow Pit(\vec{s})$

Dynamic Rules

$\neg Breezy(\vec{a})$

Prove: $\neg Safe(\vec{a}+\hat{x})$ $\quad Safe(\vec{s}) \vee Pit(\vec{s})$

$\theta = \{\vec{s}/\vec{a}+\hat{x}\}$

$Pit(\vec{a}+\hat{x})$ $\quad Breezy(\vec{s}) \vee \neg Pit(East(\vec{s}))$

$\theta = \{East(\vec{s})/\vec{a}+\hat{x}\}$

$\Rightarrow \vec{s} = West(\vec{a}+\hat{x})$

$Breezy(\vec{a})$

$North(\vec{r}, \vec{s}) \Rightarrow \vec{r} = \vec{s} + \hat{y}$

$Breezy(\vec{a}+\hat{x})$

Replace, using NSEW predicates

In CNF form

$[\neg Breezy(\vec{s}) \vee East(\vec{s},\vec{r}) \vee North(\vec{s},\vec{r}) \vee South(\vec{s},\vec{r}) \vee West(\vec{s},\vec{r})]$

$[\neg Breezy(\vec{s}) \vee Pit(\vec{r})]$

$[Breezy(\vec{s}) \vee \neg East(\vec{s},\vec{r}) \vee \neg North(\vec{s},\vec{r}) \vee \neg South(\vec{s},\vec{r}) \vee \neg West(\vec{s},\vec{r}) \vee \neg Pit(\vec{r})]$

$[\neg Pit(\vec{s}) \vee \neg Safe(\vec{s})]$

$[Pit(\vec{s}) \vee Safe(\vec{s})]$

Literal

class atom{type, val}

const/var/ hopefully not function.

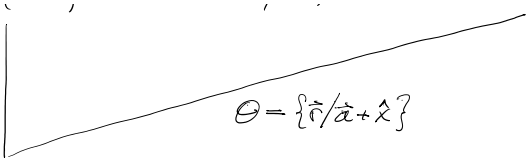
$\neg Breezy(\vec{a})$

Prove: $Safe(\vec{a}+\hat{x})$

$\neg Safe(\vec{a}+\hat{x})$ $\quad Pit(\vec{s}) \vee Safe(\vec{s})$

$\theta = \{\vec{s}/\vec{a}+\hat{x}\}$

$Pit(\vec{a}+\hat{x})$ $\quad Breezy(\vec{s}) \vee \neg East(\vec{s},\vec{r}) \vee \neg North(\vec{s},\vec{r}) \vee \neg South(\vec{s},\vec{r}) \vee \neg West(\vec{s},\vec{r}) \vee \neg Pit(\vec{r})$



$Breezy(\vec{s}) \vee \neg East(\vec{s}, \vec{a}+\hat{x}) \vee \neg North(\vec{s}, \vec{a}+\hat{x}) \vee \neg South(\vec{s}, \vec{a}+\hat{x}) \vee \neg West(\vec{s}, \vec{a}+\hat{x})$

How about a function for NSEW

$\neg Breezy(\vec{s}) \vee Pit(East(\vec{s})) \vee Pit(North(\vec{s})) \vee Pit(South(\vec{s})) \vee Pit(West(\vec{s}))$

$Breezy(\vec{s}) \vee \neg Pit(East(\vec{s}))$

$Breezy(\vec{s}) \vee \neg Pit(North(\vec{s}))$

$Breezy(\vec{s}) \vee \neg Pit(South(\vec{s}))$

$Breezy(\vec{s}) \vee \neg Pit(West(\vec{s}))$

Data structure

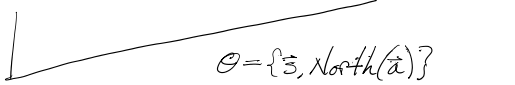
$\neg Pit(\vec{s}) \vee \neg Safe(\vec{s})$

$Pit(\vec{s}) \vee Safe(\vec{s})$ Breezy

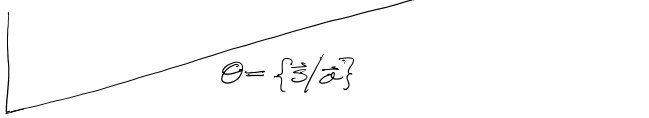
$Breezy(\vec{a})$

Prove $Safe(North(\vec{a}))$

$\Rightarrow \neg Safe(North(\vec{a}))$ $Pit(\vec{s}) \vee Safe(\vec{s})$



$Pit(North(\vec{a}))$ $Breezy(\vec{s}) \vee \neg Pit(North(\vec{s}))$



$Breezy(\vec{a})$

$\neg Pit(East(\vec{s}))$ $Pit(\vec{s})$

$\begin{pmatrix} \neg Pit \\ East(\vec{s}) \end{pmatrix}$ $\begin{pmatrix} Pit \\ \vec{s} \end{pmatrix}$

$Pred(func1(\vec{s}_1, \vec{s}_2), func2(\vec{s}_1, \vec{s}_2))$

$kb[i,j] == tuple(int, vec<tuple(int, vec<int>)>)$

Pred/neg func1 var/const.

$\{Pred1, [(func1, "var1, var2, \ldots"), (func2, "var1, var2", \ldots), \ldots]\}, \{Pred2, [(func1, "var1, var2, \ldots"), (func2, "var1, var2", \ldots), \ldots]\}$ \n

Wumpus Rules

$$Stinky(\vec{s},t) \Rightarrow Wumpus(North(\vec{s}),t) \vee Wumpus(South(\vec{s}),t) \vee Wumpus(East(\vec{s}),t) \vee Wumpus(West(\vec{s}),t)$$

$$\neg Stinky(\vec{s},t) \Rightarrow \neg Wumpus(North(\vec{s}),t) \wedge \neg Wumpus(South(\vec{s}),t) \wedge \neg Wumpus(East(\vec{s}),t) \wedge Wumpus(West(\vec{s}),t)$$

$$Scream(\vec{s},t_1) \Rightarrow \neg Wumpus(\vec{s},t_2) \wedge GreaterThan(t_2,t_1)$$

$$Wumpus(\vec{s},t) \wedge AgentAt(\vec{r},t) \wedge [SameX(\vec{r},\vec{s}) \vee SameY(\vec{r},\vec{s})] \Rightarrow Shoot(direction(\vec{r},\vec{s}))$$

We still need to determine how to incorporate walls!!

returns the direction of $\vec{r}$ from $\vec{s}$. Undef if diagonal.

Questions

- Is the current implementation "cheating" since resolution takes only one step?
- Can we ever locate pits / Wumpi exactly?
- Is it acceptable to chain inference engines?
- Does the scream predicate have an associated position?
- Does resolving the same clause twice ever give new information?