Formal Methods for Probabilistic Systems

Annabelle McIver
Carroll Morgan

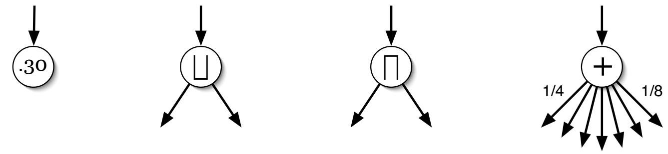
- Source-level program logic
- Meta-theorems for loops
- Examples
- Relational operational model
- Almost-certain termination
- Mu-calculus, temporal logic and games
    - Two-player probabilistic games, and their value
    - The *qM*μ and its game interpretation
    - *Minimax* and *maximin* for games
    - The denotational interpretation of *qM*μ
    - *Theorem*: the equivalence of games and denotations
    - Example: solution via *Mathematica* and *PRISM*

Two-player probabilistic games

There are two players, a *maximising* player and a *minimising* player. A turn in the game is one of the following:

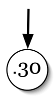
- An immediate payoff (between 0 and 1), ending the game;
- A maximising turn;
- A minimising turn; or
- A probabilistic choice.

The maximising player strives to make the (expected) payoff as high as possible; the minimising player tries to make it as low as possible. Neither player has any control over probabilistic outcomes.

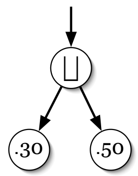


The "value" of a game — examples

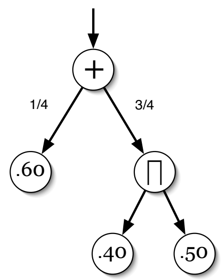
This game has value .30



This game has value .50

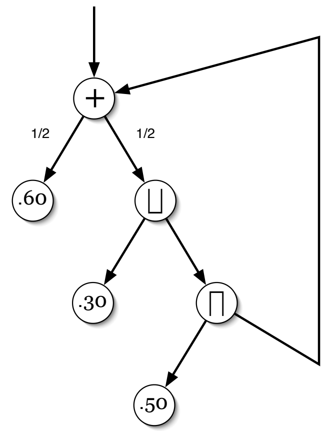


This game has value .45



The "value" of a game

This game has value .55



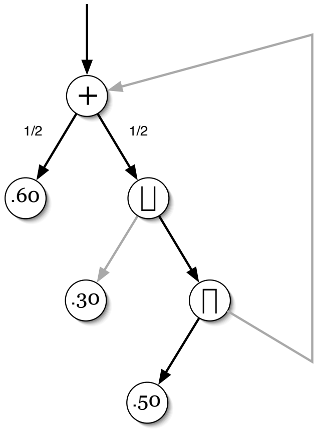
The "value" of a game

This game has value .55

The game's value is the ***greatest*** return that *Max* can force *no matter what* Min *does*.

It is also is the ***least*** return that *Min* can force *no matter what* Max *does*.

That these are the same *must be proved*.



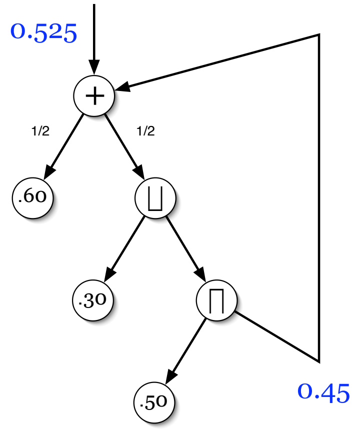
Max uses the strategy "go right"; *Min* uses the strategy "go left".

The "value" of a game

This game has value .55

0.0
0.45
0.525
0.55
0.55

Iterate to a solution

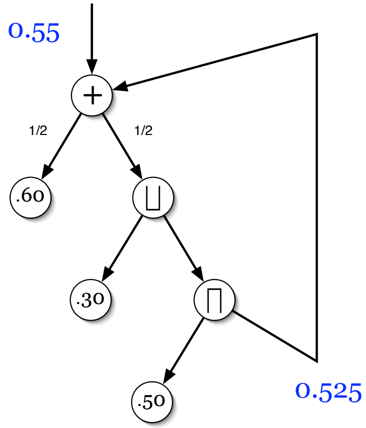


The "value" of a game

This game has value .55

0.0
0.45
0.525
0.55
0.55

Iterate to a solution



A logic for two-player probabilistic games

The two-player games are formalised by a *quantitative modal-mu calculus* logic (extending Kozen). The principal theorem we prove is that the *value of a formula* can be determined in either of two equivalent ways:

- Use the formula, à la Stirling (but extended by us), to play a probabilistic minimax-over-strategies game as above. Operational reasoning is used.
- Interpret the formula, à la Kozen (but extended by us), denotationally in a lattice of real-valued functions. Least- and greatest fixed-points are used.

The equivalence means that we can reason operationally about whether a formula is appropriate for our application (Stirling), and then use mathematical semantics to manipulate it (Kozen).

C. Stirling. Local model-checking games. *CONCUR '95*. LNCS 962, 1-11, 1995.
D. Kozen. Results on the propositional mu-calculus. *TCS* 27, 333-54, 1983.

The quantitative modal mu-calculus qMμ

We operate over a state space S (usually countable, often finite), and a derived space $\mathcal{R}.S$ of probabilistic/demonic transitions over S in which we can express the tree-building nodes we saw earlier.

$$\begin{array}{rcl} \phi & \hat{=} & X \mid \mathsf{A} \mid \{\mathsf{k}\}\phi \\ & \mid & \phi_1 \sqcap \phi_2 \mid \phi_1 \sqcup \phi_2 \mid \phi_1 \lhd \mathsf{G} \rhd \phi_2 \\ & \mid & (\mu X \cdot \phi) \mid (\nu X \cdot \phi) \end{array}$$

- Variables X are of type $S \to [0,1]$, and are used for binding fixed points.
- Terms A stand for fixed functions in $S \to [0,1]$.
- Terms k represent probabilistic state-to-state transitions in $\mathcal{R}.S$.
- Terms G describe Boolean functions of S, used in $\lhd$ ("if") G $\rhd$ ("else") style.

The tree-building transitions

We shall assume generally that S is a countable state space (though for the principal result we restrict to finiteness). If f is a function with domain X then by $f.x$ we mean f applied to x, and $f.x.y$ is $(f.x).y$ where appropriate; functional composition is written with $\circ$, so that $(f{\circ}g).x = f.(g.x)$.

We denote the set of discrete probability *sub*-distributions over a set X by $\overline{X}$: it is the set of functions from X into the real interval $[0,1]$ that sum to *no more than* one.

If A is a random variable with respect to some probability space, and δ is some probability sub-distribution, we write $\int_\delta A$ for the expected value of A with respect to δ.

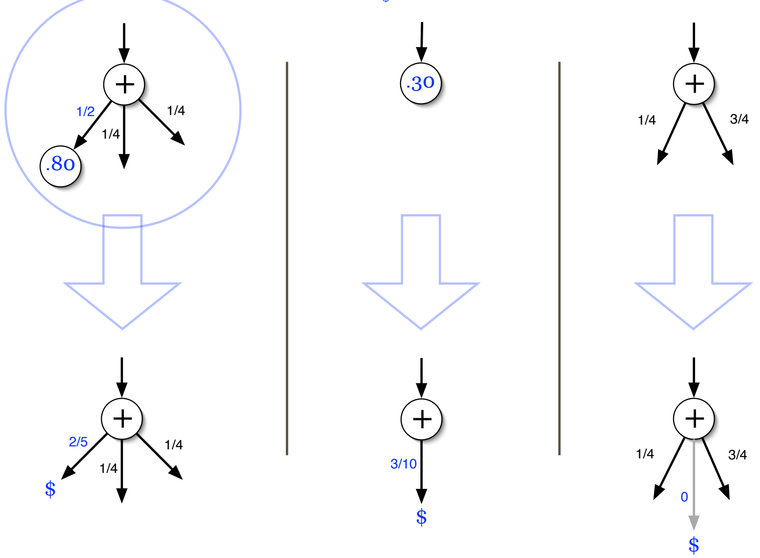
The space of generalised probabilistic transitions $\mathcal{R}.S$ comprises the functions t in $S \to \overline{S_\$}$ where $S_\$$ is just the state space S with a special "payoff" state \$ adjoined.

Thus $\overline{S_\$}$ is the set of sub-distributions over that, so that the elements t of $\mathcal{R}.S$ give the probability of passage from initial s to final (proper) s' as $t.s.s'$; any deficit $1 - \sum_{s'} t.s.s'$ is interpreted as the probability of an immediate halt with payoff

$$t.s.\$/(1 - \sum_{s':S} t.s.s') \ .$$

The tree-building transitions — a coding trick

$$\mathcal{R}.S \quad \hat{=} \quad S \to \overline{S_\$}$$



From a formula to a game

The game is between two players *Max* and *Min*. Play progresses through a sequence of *game positions*, each of which is either a pair (ϕ, s) where ϕ is a formula and s is a state in S, or a single (y) for some real-valued payoff y in $[0,1]$. We use "colours" to handle repeated returns to a fixed point.

A sequence of game positions is called a *game path* and is of the form (ϕ_0, s_0), $(\phi_1, s_1), \ldots$ with (if finite) a payoff position (y) at the end. The initial formula ϕ_0 is the given ϕ, and s_0 is an *initial* state in S. A move from position (ϕ_i, s_i) to (ϕ_{i+1}, s_{i+1}) or to (y) is specified by the following rules.

From a formula to a game

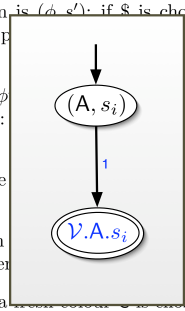
If the current game position is (ϕ_i, s_i), then play proceeds as follows:

1. Free variables X do not occur in the game — their role is taken over by “colours”.
2. If ϕ_i is A then the game terminates in position (y) where $y = \mathcal{V}.\mathsf{A}.s_i$.
3. If ϕ_i is $\{\mathsf{k}\}\phi$ then the distribution $\mathcal{V}.\mathsf{k}.s_i$ is used to choose either a next state s' in S or possibly the payoff state \$. If a state s' is chosen, then the next game position is (ϕ, s'); if \$ is chosen, then the next position is (y), where y is the payoff $\mathcal{V}.\mathsf{k}.s.\$/(1 - \sum_{s':S} \mathcal{V}.\mathsf{k}.s.s')$, and the game terminates.
4. If ϕ_i is $\phi' \sqcap \phi''$ (resp. $\phi' \sqcup \phi''$) then *Min* (resp. *Max*) chooses one of the minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen ’junct ϕ' or ϕ''.
5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

From a formula to a game

If the current game position is (ϕ_i, s_i), then play proceeds as follows:

1. Free variables X do not occur in the game — their role is taken over by “colours”.
2. If ϕ_i is A then the game terminates in position (y) where $y = \mathcal{V}.\mathsf{A}.s_i$.
3. If ϕ_i is $\{\mathsf{k}\}\phi$ then the distribution $\mathcal{V}.\mathsf{k}.s_i$ is used to choose either a next state s' in S or possibly the payoff state \$. If a state s' is chosen, then the next game position … chosen, then the next position is (y), where y is the p… $\mathcal{V}.\mathsf{k}.s.s')$, and the game terminates.

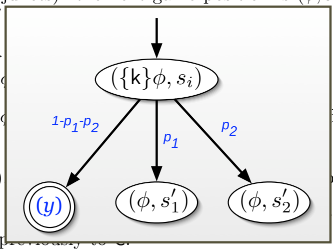


4. If ϕ_i is $\phi' \sqcap \phi''$ (resp. ϕ… …sp. *Max*) chooses one of the minjuncts (maxjuncts): …ion is (ϕ, s_i), where ϕ is the chosen ’junct ϕ' or ϕ''.
5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the… …s (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then… …chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later… …position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a… …en and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

From a formula to a game

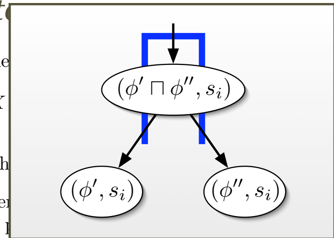
If the current game position is (ϕ_i, s_i), then play proceeds as follows:

1. Free variables X do not occur in the game — their role is taken over by “colours”.
2. If ϕ_i is A then the game terminates in position (y) where $y = \mathcal{V}.\mathsf{A}.s_i$.
3. If ϕ_i is $\{\mathsf{k}\}\phi$ then the distribution $\mathcal{V}.\mathsf{k}.s_i$ is used to choose either a next state s' in S or possibly the payoff state \$. If a state s' is chosen, then the next game position is (ϕ, s'); if \$ is chosen, then the next position is (y), where y is the payoff $\mathcal{V}.\mathsf{k}.s.\$/(1 - \sum_{s':S} \mathcal{V}.\mathsf{k}.s.s')$, and the game terminates.
4. If ϕ_i is $\phi' \sqcap \phi''$ (resp. $\phi' \sqcup \phi''$) then *Min* (resp. *Max*) chooses one of the minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen ’junct ϕ'…



5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd$… $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (…
6. If ϕ_i is $(\mu X \cdot$… …l is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$… (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$… …und as for μ.
8. If ϕ_i is a colour… …$_i)$, where Φ is the formula bound p…

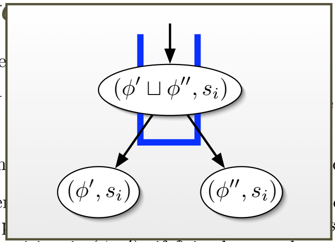
From a formula t…



If the current game… …as follows:

1. Free variables X… …is taken over by “colours”.
2. If ϕ_i is A then th… …e $y = \mathcal{V}.\mathsf{A}.s_i$.
3. If ϕ_i is $\{\mathsf{k}\}\phi$ the… …ose either a next state s' in S or… …$'$ is chosen, then the next game position is (ϕ, s'); if \$ is chosen, then the next position is (y), where y is the payoff $\mathcal{V}.\mathsf{k}.s.\$/(1 - \sum_{s':S} \mathcal{V}.\mathsf{k}.s.s')$, and the game terminates.
4. If ϕ_i is $\phi' \sqcap \phi''$ (resp. $\phi' \sqcup \phi''$) then *Min* (resp. *Max*) chooses one of the minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen ’junct ϕ' or ϕ''.
5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

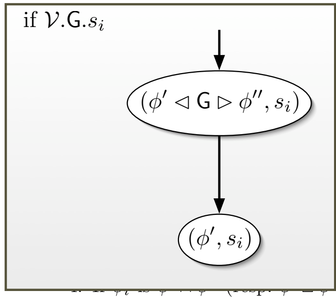
From a formula t

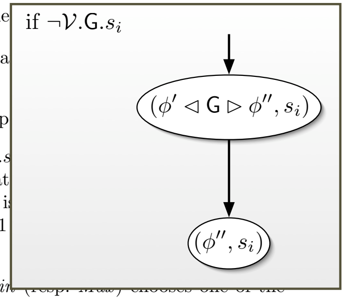


If the current game … as follows:

1. Free variables X … is taken over by "colours".
2. If ϕ_i is A then th… $y = \mathcal{V}.\mathsf{A}.s_i$.
3. If ϕ_i is $\{\mathsf{k}\}\phi$ the… ose either a next state s' in S or … s' is chosen, then the next game position is (ϕ, s'); if \$ is chosen, then the next position is (y), where y is the payoff $\mathcal{V}.\mathsf{k}.s.\$/(1 - \sum_{s':S} \mathcal{V}.\mathsf{k}.s.s')$, and the game terminates.
4. If ϕ_i is $\phi' \sqcap \phi''$ (resp. $\phi' \sqcup \phi''$) then *Min* (resp. *Max*) chooses one of the minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen 'junct ϕ' or ϕ''.
5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

From a formula to a game



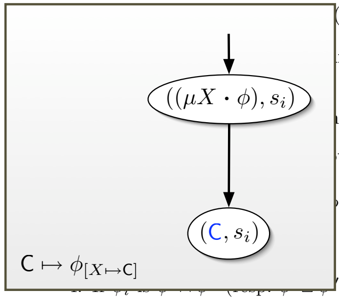


(ϕ_i, s_i), the
r in the ga
inates in p
ution $\mathcal{V}.\mathsf{k}.s$
payoff stat
$, s')$; if \$ i
$\mathcal{V}.\mathsf{k}.s.\$/(1$

…') then *Mi*… minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen 'junct ϕ' or ϕ''.

5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

From a formula to a game

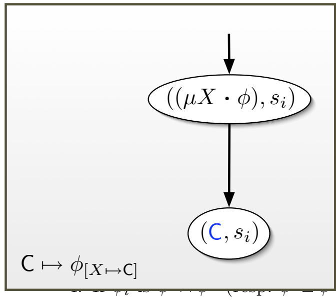


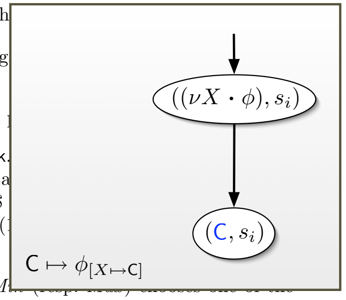
(ϕ_i, s_i), then play proceeds as follows:
r in the game — their role is taken over by
inates in position (y) where $y = \mathcal{V}.\mathsf{A}.s_i$.
ution $\mathcal{V}.\mathsf{k}.s_i$ is used to choose either a next
payoff state \$. If a state s' is chosen, then
$, s')$; if \$ is chosen, then the next position
$\mathcal{V}.\mathsf{k}.s.\$/(1 - \sum_{s':S} \mathcal{V}.\mathsf{k}.s.s')$, and the game

…') then *Min* (resp. *Max*) chooses one of the minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen 'junct ϕ' or ϕ''.

5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

From a formula to a game



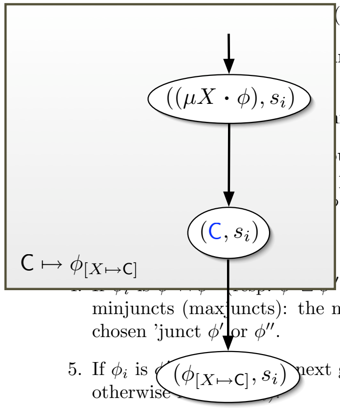


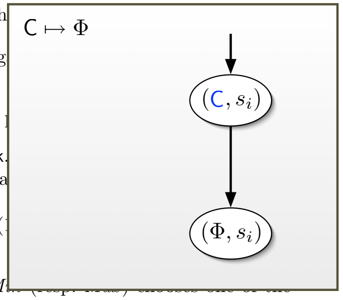
(ϕ_i, s_i), th
r in the g
inates in
ution $\mathcal{V}.\mathsf{k}.$
payoff sta
$, s')$; if \$
$\mathcal{V}.\mathsf{k}.s.\$/($

…') then *M*… minjuncts (maxjuncts): the next game position is (ϕ, s_i), where ϕ is the chosen 'junct ϕ' or ϕ''.

5. If ϕ_i is $\phi' \lhd \mathsf{G} \rhd \phi''$, the next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise it is (ϕ'', s_i).
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).
7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.
8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

From a formula to a game





(ϕ_i, s_i), th

r in the g

inates in

ution $\mathcal{V}.\mathsf{k}.$

payoff sta

$, s')$; if \$

$\mathcal{V}.\mathsf{k}.s.\$/($

) then M

minjuncts (max juncts): the next game position is (ϕ, s_i), where ϕ is the chosen 'junct ϕ' or ϕ''.

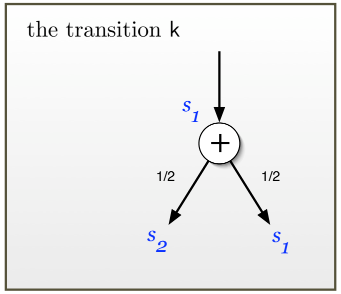
5. If ϕ_i is next game position is (ϕ', s_i) if $\mathcal{V}.\mathsf{G}.s_i$ holds, and otherwise

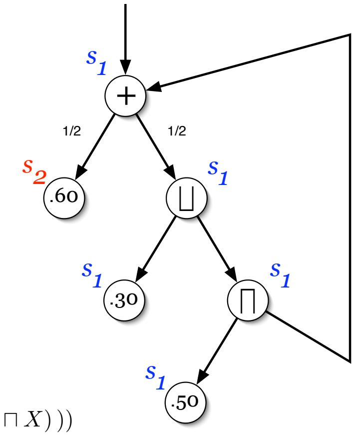
6. If ϕ_i is $(\mu X \cdot \phi)$ then a fresh colour C is chosen and is bound to the formula $\phi_{[X \mapsto \mathsf{C}]}$ for later use; the next game position is (C, s_i).

7. If ϕ_i is $(\nu X \cdot \phi)$, then a fresh colour C is chosen and bound as for μ.

8. If ϕ_i is a colour C, then the next game position is (Φ, s_i), where Φ is the formula bound previously to C.

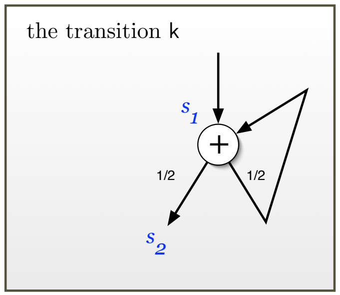
From a game to a formula

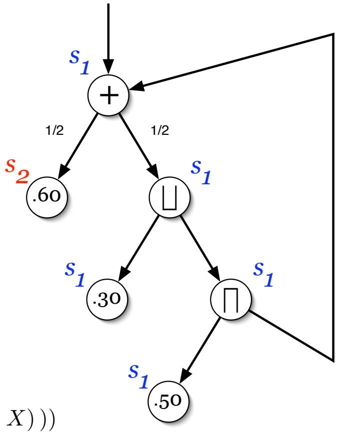




$(\mu X \cdot \{\mathsf{k}\}(\,(.60) \lhd \text{“in } s_2\text{”} \rhd (.30) \sqcup ((.50) \sqcap X)\,))$
$(\nu X \cdot \{\mathsf{k}\}(\,(.60) \lhd \text{“in } s_2\text{”} \rhd (.30) \sqcup ((.50) \sqcap X)\,))$

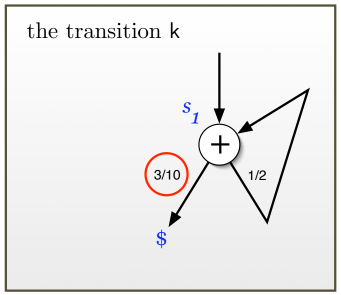
From a game to a formula

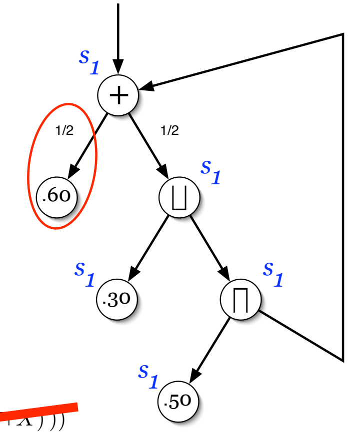




$(\mu X \cdot \{\mathsf{k}\}(\,(.60) \lhd \text{“in } s_2\text{”} \rhd (.30) \sqcup ((.50) \sqcap X)\,))$
$(\nu X \cdot \{\mathsf{k}\}(\,(.60) \lhd \text{“in } s_2\text{”} \rhd (.30) \sqcup ((.50) \sqcap X)\,))$

From a game to a formula

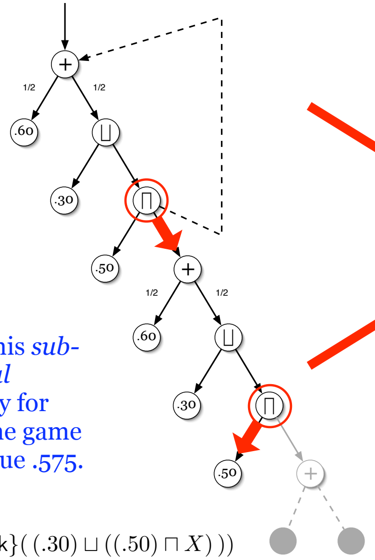


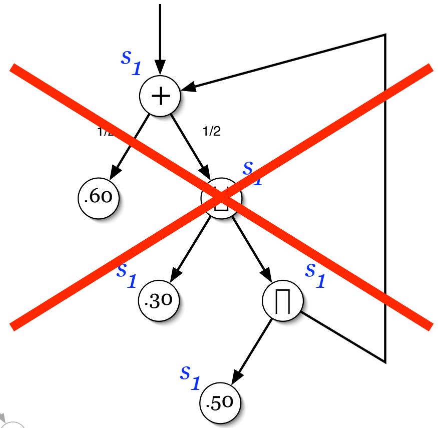


$\cancel{(\mu X \cdot \{\mathsf{k}\}(\,(.60) \lhd \text{“in } s_2\text{”} \rhd (.30) \sqcup ((.50) \sqcap X)\,))}$
$\cancel{(\nu X \cdot \{\mathsf{k}\}(\,(.60) \lhd \text{“in } s_2\text{”} \rhd (.30) \sqcup ((.50) \sqcap X)\,))}$

$(\mu X \cdot \{\mathsf{k}\}(\,(.30) \sqcup ((.50) \sqcap X)\,))$

Do strategies have memory?

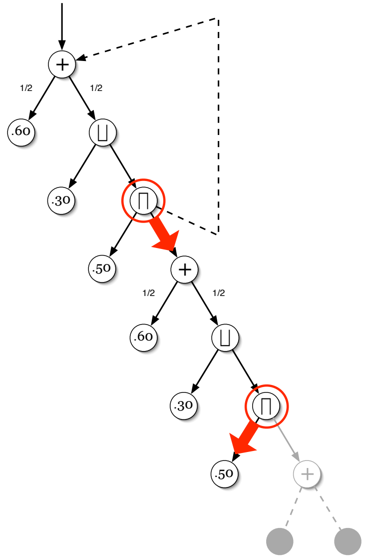




With this *sub-optimal* strategy for *Min*, the game has value .575.

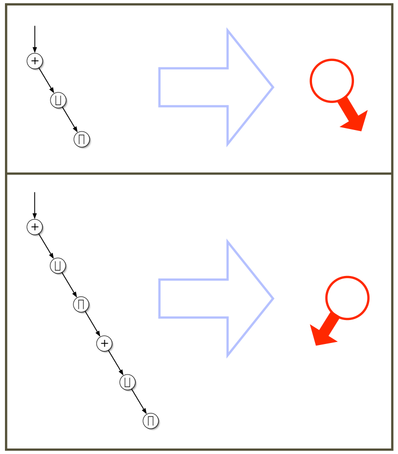
$(\mu X \cdot \{\mathsf{k}\}(\,(.30) \sqcup ((.50) \sqcap X)\,))$

Reasoning about strategies

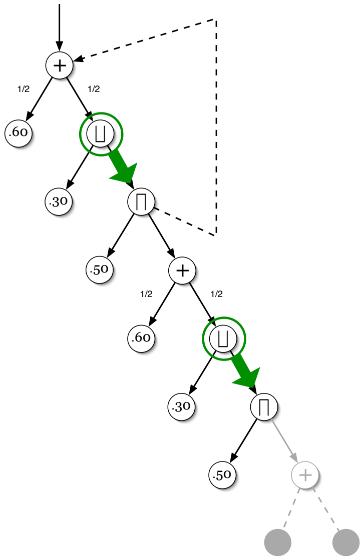


A *strategy* is a function from game-paths to Boolean, *decided in advance*.

A portion of *Min*'s strategy.

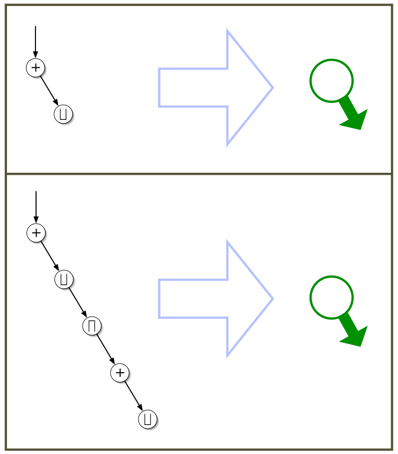


Reasoning about strategies

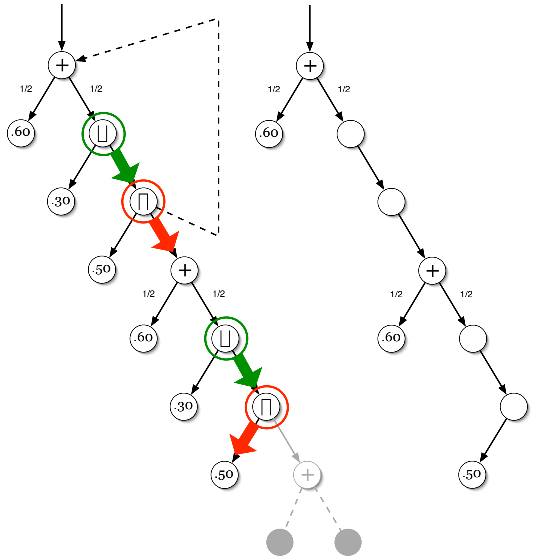


A *strategy* is a function from game-paths to Boolean, *decided in advance*.

A portion of *Max*'s strategy.



Reasoning about strategies



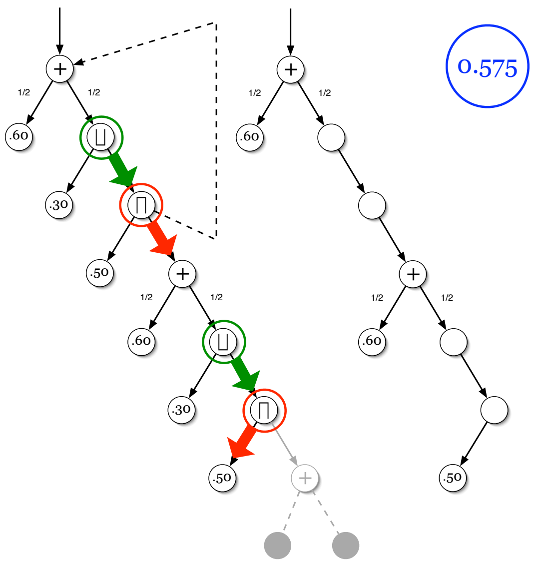
When the strategies are applied, the "reduced" game-tree is purely probabilistic, and has a value.

In this case, the value is

$$1/2 * .60 \\ + 1/2 * (1/2 * .60 + \\ 1/2 * .50)\,,$$

that is 0.575.

Reasoning about strategies

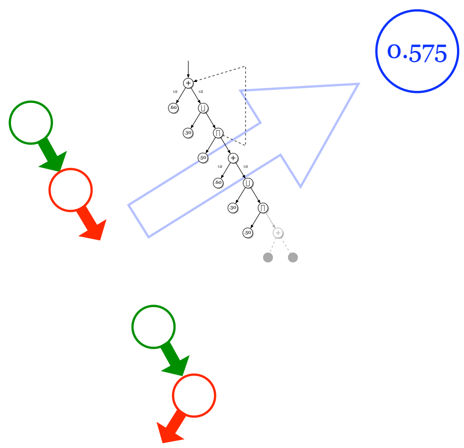


Thus a “full” game, with its maximising and minimising nodes, is a function

from strategy-pairs (one for *Max* and one for *Min*)

to [0,1] (which is the value of the reduced, purely probabilistic game remaining after the strategies have been applied).

Reasoning about strategies



Thus a “full” game, with its maximising and minimising nodes, is a function

from strategy-pairs (one for *Max* and one for *Min*)

to [0,1] (which is the value of the reduced, purely probabilistic game remaining after the strategies have been applied).

Reasoning about strategies

Thus a “full” game, with its maximising and minimising nodes, is a function from strategy-pairs (one for *Max* and one for *Min*) to [0,1] (which is the value of the reduced, purely probabilistic game remaining after the strategies have been applied).

$\overline{\sigma}$ is the maximising strategy $\qquad$ $\underline{\sigma}$ is the minimising strategy

The value of a game played from formula ϕ and initial state s, with fixed strategies $\underline{\sigma}, \overline{\sigma}$, is given by the expected value

$$\int_{[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s} Val$$

generates the purely-probabilistic game-tree for the given strategies, formula and initial state

of *Val* over the (probability distribution determined by the) game-tree $[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s$ generated by the formula, the strategies and the initial state.

Reasoning about strategies: the game is well defined

$$\boxed{\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s} Val \quad = \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s} Val}$$

$\overline{\sigma}$ is the maximising strategy $\qquad$ $\underline{\sigma}$ is the minimising strategy

The value of a game played from formula ϕ and initial state s, with fixed strategies $\underline{\sigma}, \overline{\sigma}$, is given by the expected value

$$\int_{[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s} Val$$

generates the purely-probabilistic game-tree for the given strategies, formula and initial state

of *Val* over the (probability distribution determined by the) game-tree $[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s$ generated by the formula, the strategies and the initial state.

Reasoning about strategies: the game is well defined

$$\boxed{\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val} = \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

$\overline{\sigma}$ is the maximising strategy $\qquad$ $\underline{\sigma}$ is the minimising strategy

It doesn't matter who "goes first" in selecting an overall strategy.

On the left, for any strategy *Min* might select, we allow *Max* subsequently to choose the best "counter-strategy" from his point of view. The best *Min* can do under these circumstances is the *minimax* value of the game.

Reasoning about strategies: the game is well defined

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val = \boxed{\sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val}$$

$\overline{\sigma}$ is the maximising strategy $\qquad$ $\underline{\sigma}$ is the minimising strategy

It doesn't matter who "goes first" in selecting an overall strategy.

On the left, for any strategy *Min* might select, we allow *Max* subsequently to choose the best "counter-strategy" from his point of view. The best *Min* can do under these circumstances is the *minimax* value of the game.

On the right we have the *maximin* value, where *Max* goes first.

Reasoning about strategies: the game is well defined

$$\boxed{\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val = \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val}$$

$\overline{\sigma}$ is the maximising strategy $\qquad$ $\underline{\sigma}$ is the minimising strategy

It doesn't matter who "goes first" in selecting an overall strategy.

This is a corollary of our main result — the equivalence of the game interpretation and the denotational interpretation (next!) of quantitative modal mu-calculus formulae.
It was however proved in its own right by Martin, in the closely related context of perfect-information stochastic parity-games.

DA Martin. The determinacy of Blackwell Games. *Jnl. Sym. Log. 63(4)*:1565-81, 1998.

The game is well defined...

$$\boxed{\begin{array}{c} \sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val = \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val \\ = \|\phi\|_{\mathcal{V}} \end{array}}$$

... *and is equal to the denotational interpretation of the formula that generated the game.*

The denotational interpretation of qMμ

We operate over a state space S (usually countable, often finite), and a derived space $\mathcal{R}.S$ of probabilistic/demonic transitions over S in which we can express the tree-building nodes we saw earlier.

$$
\begin{array}{lll}
\phi & \hat{=} & X \mid \mathsf{A} \mid \{\mathsf{k}\}\phi \\
 & \mid & \phi_1 \sqcap \phi_2 \mid \phi_1 \sqcup \phi_2 \mid \phi_1 \lhd \mathsf{G} \rhd \phi_2 \\
 & \mid & (\mu X \cdot \phi) \mid (\nu X \cdot \phi)
\end{array}
$$

- Variables X are of type $S \to [0,1]$, and are used for binding fixed points.
- Terms A stand for fixed functions in $S \to [0,1]$.
- Terms k represent probabilistic state-to-state transitions in $\mathcal{R}.S$.
- Terms G describe Boolean functions of S, used in $\lhd$ ("if") G $\rhd$ ("else") style.

The denotational interpretation of qMμ

1. $\|X\|_{\mathcal{V}} \quad \hat{=} \quad \mathcal{V}.X$.
2. $\|\mathsf{A}\|_{\mathcal{V}} \quad \hat{=} \quad \mathcal{V}.\mathsf{A}$.
3. $\|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \quad \hat{=} \quad \mathcal{V}.\mathsf{k}.s.\$ + \int_{\mathcal{V}.\mathsf{k}.s} \|\phi\|_{\mathcal{V}}$.
4. $\|\phi' \sqcap \phi''\|_{\mathcal{V}}.s \quad \hat{=} \quad \|\phi'\|_{\mathcal{V}}.s \ \mathsf{min} \ \|\phi''\|_{\mathcal{V}}.s$; and
   $\|\phi' \sqcup \phi''\|_{\mathcal{V}}.s \quad \hat{=} \quad \|\phi'\|_{\mathcal{V}}.s \ \mathsf{max} \ \|\phi''\|_{\mathcal{V}}.s$.
5. $\|\phi' \lhd \mathsf{G} \rhd \phi''\|_{\mathcal{V}}.s \quad \hat{=} \quad \|\phi'\|_{\mathcal{V}}.s \ \underline{\text{if}} \ (\mathcal{V}.\mathsf{G}.s) \ \underline{\text{else}} \ \|\phi''\|_{\mathcal{V}}.s$.
6. $\|(\mu X \cdot \phi)\|_{\mathcal{V}} \quad \hat{=} \quad (\mathsf{lfp}\, x \cdot \|\phi\|_{\mathcal{V}_{[X\mapsto x]}})$ where by $(\mathsf{lfp}\, x \cdot exp)$ we mean the least fixed-point of the function $(\lambda x \cdot exp)$.
7. $\|(\nu X \cdot \phi)\|_{\mathcal{V}} \quad \hat{=} \quad (\mathsf{gfp}\, x \cdot \|\phi\|_{\mathcal{V}_{[X\mapsto x]}})$.

$\phi \ \hat{=} \ X \mid \mathsf{A} \mid \{\mathsf{k}\}\phi \mid \phi_1 \sqcap \phi_2 \mid \phi_1 \sqcup \phi_2 \mid \phi_1 \lhd \mathsf{G} \rhd \phi_2 \mid (\mu X \cdot \phi) \mid (\nu X \cdot \phi)$

The denotational interpretation of qMμ

1. $\|X\|_{\mathcal{V}} \quad \hat{=} \quad \mathcal{V}.X$.
2. $\|\mathsf{A}\|_{\mathcal{V}} \quad \hat{=} \quad \mathcal{V}.\mathsf{A}$.

$$\|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \quad \hat{=} \quad \mathcal{V}.\mathsf{k}.s.\$ + \int_{\mathcal{V}.\mathsf{k}.s} \|\phi\|_{\mathcal{V}}$$

4. $\|\phi' \sqcap \phi''\|_{\mathcal{V}}.s \quad \hat{=} \quad \|\phi'\|_{\mathcal{V}}.s \ \mathsf{min} \ \|\phi''\|_{\mathcal{V}}.s$; and
   $\|\phi' \sqcup \phi''\|_{\mathcal{V}}.s \quad \hat{=} \quad \|\phi'\|_{\mathcal{V}}.s \ \mathsf{max} \ \|\phi''\|_{\mathcal{V}}.s$.
5. $\|\phi' \lhd \mathsf{G} \rhd \phi''\|_{\mathcal{V}}.s \quad \hat{=} \quad \|\phi'\|_{\mathcal{V}}.s \ \underline{\text{if}} \ (\mathcal{V}.\mathsf{G}.s) \ \underline{\text{else}} \ \|\phi''\|_{\mathcal{V}}.s$.
6. $\|(\mu X \cdot \phi)\|_{\mathcal{V}} \quad \hat{=} \quad (\mathsf{lfp}\, x \cdot \|\phi\|_{\mathcal{V}_{[X\mapsto x]}})$ where by $(\mathsf{lfp}\, x \cdot exp)$ we mean the least fixed-point of the function $(\lambda x \cdot exp)$.
7. $\|(\nu X \cdot \phi)\|_{\mathcal{V}} \quad \hat{=} \quad (\mathsf{gfp}\, x \cdot \|\phi\|_{\mathcal{V}_{[X\mapsto x]}})$.

The denotational interpretation of qMμ

$$\|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \quad \hat{=} \quad \mathcal{V}.\mathsf{k}.s.\$ + \int_{\mathcal{V}.\mathsf{k}.s} \|\phi\|_{\mathcal{V}}$$

Denotations $\|\phi\|_{\mathcal{V}}$ are random variables over S, that is of type $S \to [0,1]$. Thus the effect of k — the function $\|\mathsf{k}\|_{\mathcal{V}}$ so to speak — is to transform one of these random variables into another. It can be considered to be of type

$$(S \to [0,1]) \quad \to \quad (S \to [0,1]) \ .$$

The random variables are a complete lattice with the order (pointwise-extended) $\leq$ and so have least/greatest element the everywhere-zero/everywhere-one function respectively.

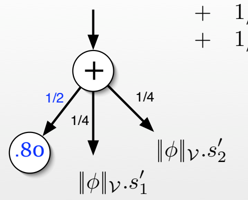
Least and greatest fixed points are taken within this lattice.

The denotational interpretation of qMμ

$$\|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \quad \hat{=} \quad \mathcal{V}.\mathsf{k}.s.\$ \; + \; \int_{\mathcal{V}.\mathsf{k}.s} \|\phi\|_{\mathcal{V}}$$

the transition k

$$\begin{aligned} \|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \; = \; & \; 2/5 \\ + & \; 1/4 \times \|\phi\|_{\mathcal{V}}.s'_1 \\ + & \; 1/4 \times \|\phi\|_{\mathcal{V}}.s'_2 \end{aligned}$$

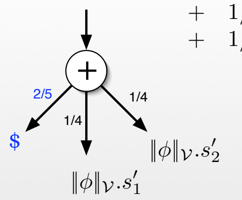


The denotational interpretation of qMμ

$$\|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \quad \hat{=} \quad \mathcal{V}.\mathsf{k}.s.\$ \; + \; \int_{\mathcal{V}.\mathsf{k}.s} \|\phi\|_{\mathcal{V}}$$

the transition k

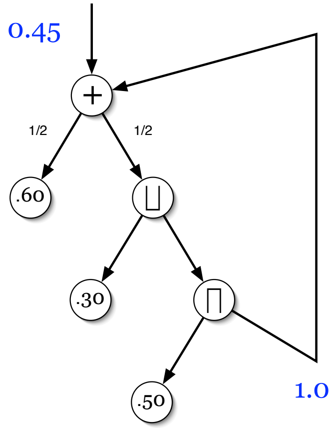
$$\begin{aligned} \|\{\mathsf{k}\}\phi\|_{\mathcal{V}}.s \; = \; & \; 2/5 \\ + & \; 1/4 \times \|\phi\|_{\mathcal{V}}.s'_1 \\ + & \; 1/4 \times \|\phi\|_{\mathcal{V}}.s'_2 \end{aligned}$$



The denotational interpretation of qMμ

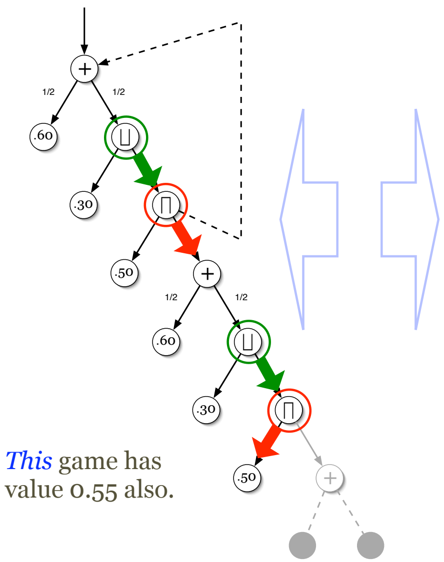
0.0	1.0
0.45	0.45
0.525	0.525
0.55	0.55
0.55	0.55

Least and greatest fixed-points are the same, in this example.



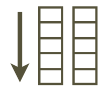
This game has value 0.55.

The two equivalent interpretations of qMμ



This game has value 0.55 also.

$$(\mu X \cdot \{\mathsf{k}\}(\,(.30) \sqcup ((.50) \sqcap X)\,))$$



This game has value 0.55.

Proof of equivalence

Note that we do not assume the minimax/maximin value of the game is well defined; that is a corollary of the proof to come.

Step 1: We augment the denotational semantics so that it too takes two strategy arguments, and show *equivalence* of the game- and denotational interpretations *when the strategies are fixed*.

> For all closed $qM\mu$ formulae ϕ, valuations $\mathcal{V}$, states s and strategies $\underline{\sigma}, \overline{\sigma}$, we have
> $$\int_{[\![\phi]\!]^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s} \mathit{Val} \quad = \quad \|\phi\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.s \ .$$

The argument uses routine denotational techniques — structural induction *etc.* — and is detailed only in the case of fixed points.

Proof of equivalence (Step 1)

Extended denotational semantics

1. $\|X\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}} \quad \hat{=} \quad \mathcal{V}.X \ .$
2. $\|\mathsf{A}\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi.s \quad \hat{=} \quad \mathcal{V}.\mathsf{A}.s \ .$
3. $\|\{\mathsf{k}\}\Phi\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi.s \quad \hat{=} \quad \mathcal{V}.\mathsf{k}.s.\$ \ + \ \int_{\mathcal{V}.\mathsf{k}.s} \|\Phi\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi^+ \ .$

4. $\|\phi' \sqcap \phi''\|_{\mathcal{V}}.s \ \hat{=} \ \|\phi'\|_{\mathcal{V}}.s \ \mathsf{min} \ \|\phi''\|_{\mathcal{V}}.s$; and
$\|\phi' \sqcup \phi''\|_{\mathcal{V}}.s \ \hat{=} \ \|\phi'\|_{\mathcal{V}}.s \ \mathsf{max} \ \|\phi''\|_{\mathcal{V}}.s$.

4. $\|\Phi' \sqcap \Phi''\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi \quad \hat{=} \quad \begin{array}{ll} \|\Phi'\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi^+ & \text{if } \underline{\sigma}.\pi^+ \\ \|\Phi''\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi^+ & \text{otherwise.} \end{array}$
5. $\|\Phi' \lhd \mathsf{G} \rhd \Phi''\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi.s \quad \hat{=} \quad \begin{array}{ll} \|\Phi'\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi^+.s & \text{if } \mathcal{V}.\mathsf{G}.s \\ \|\Phi''\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi^+.s & \text{otherwise.} \end{array}$
6. $\|(\mu X \cdot \Phi)\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi \quad \hat{=} \quad (\mathsf{lfp}\, x \cdot \|\Phi\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}_{[X\mapsto x]}}).\pi^+$
7. $\|(\nu X \cdot \Phi)\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}}.\pi \quad \hat{=} \quad (\mathsf{gfp}\, x \cdot \|\Phi\|^{\underline{\sigma},\overline{\sigma}}_{\mathcal{V}_{[X\mapsto x]}}).\pi^+$
8. (Colours are not used in $\|\cdot\|$ semantics.)

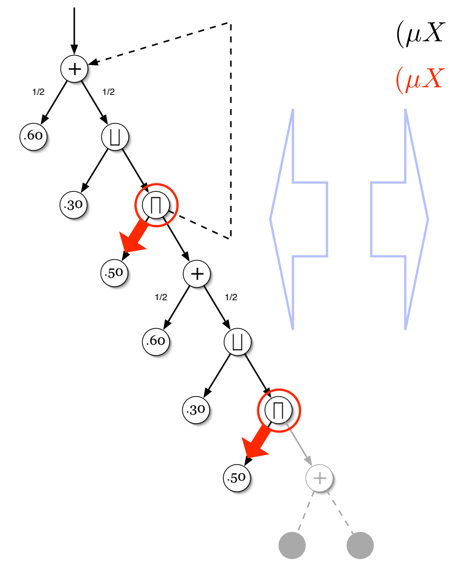
Proof of equivalence (Step 2)

Step 2: We show that *memoriless strategies suffice* in the denotational interpretation. (For this we must restrict to finite state-spaces.)

> For any formula ϕ, possibly containing strategy operators $\sqcap/\sqcup$, and valuation $\mathcal{V}$, there are state-predicate tuples $\underline{\mathsf{G}}/\overline{\mathsf{G}}$ — possibly depending on $\mathcal{V}$ — such that
> $$\|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}} \quad = \quad \|\phi\|_{\mathcal{V}} \quad = \quad \|\phi_{\overline{\mathsf{G}}}\|_{\mathcal{V}} \ .$$

We are stating that for any formula (containing maximum and minimum operators), there are two denotationally equivalent formulae: one in which all minima have been replaced by Boolean choices (but leaving the maxima); and another in which all maxima have been replaced by Boolean choices (but leaving the minima).

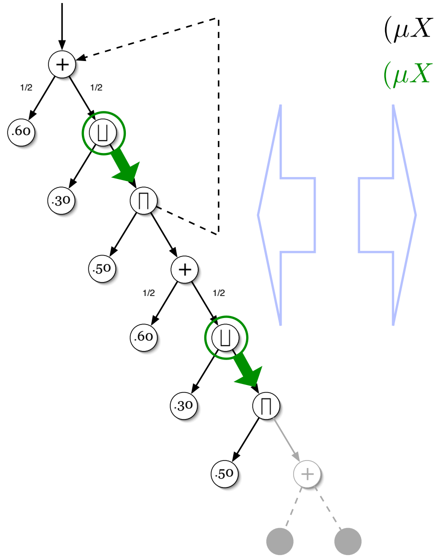
Proof of equivalence (Step 2)



$(\mu X \cdot \{\mathsf{k}\}(\,(.30) \sqcup ((.50) \sqcap X)\,))$

$(\mu X \cdot \{\mathsf{k}\}(\,(.30) \sqcup (.50 \lhd \mathsf{true} \rhd X)))$

Proof of equivalence (Step 2)



$$(\mu X \cdot \{\mathsf{k}\}(\,(.30) \sqcup ((.50) \sqcap X)\,))$$

$$(\mu X \cdot \{\mathsf{k}\}((.30) \lhd \mathsf{false} \rhd (.50) \sqcap X))$$

Proof of equivalence (Step 2)

The argument for Step 2 is straightforward except in the case of maxima within least fixed-points (or minima within greatest fixed-points).

For those cases a considerably more involved analytical argument is used, based on techniques employed by Everett.

H Everett. *Recursive games*. In *Contributions to the Theory of Games III*, Vol. 39 of *Ann. Math. Stud.*, 47-78, Princeton University Press, 1957.

Proof of equivalence (Step 3)

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val \quad \stackrel{?}{=} \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

4 5

Step 3: We use Step 2 to show that the *minimax and maximin* denotational interpretations "bracket" the strategy-free denotational interpretation.

For all $qM\mu$ formulae ϕ, valuations $\mathcal{V}$ and strategies $\underline{\sigma}, \overline{\sigma}$, we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \|\phi\|_{\mathcal{V}} \quad \leq \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \;.$$

1 2 3

In fact, all five expressions on this page are equal.

The equality of 4 and 5 is the corollary we mentioned earlier.

Proof of equivalence (Step 3)

For all $qM\mu$ formulae ϕ, valuations $\mathcal{V}$ and strategies $\underline{\sigma}, \overline{\sigma}$, we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \|\phi\|_{\mathcal{V}} \quad \leq \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \;.$$

Proof of Step 3 (using Step 2)

Step 2 provides us with predicates $\overline{\mathsf{G}}$ and $\underline{\mathsf{G}}$ satisfying

$$\|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}} \quad = \quad \|\phi\|_{\mathcal{V}} \quad = \quad \|\phi_{\overline{\mathsf{G}}}\|_{\mathcal{V}} \;.$$

We start from the *lhs* and observe that

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \sqcup_{\overline{\sigma}} \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}}^{\overline{\sigma}} \;,$$

where on the right we omit the now-ignored $\underline{\sigma}$ argument — because the $\sqcap_{\underline{\sigma}}$ can select exactly the predicates provided by $\underline{\mathsf{G}}$ simply by making an appropriate choice of $\underline{\sigma}$.

Proof of equivalence (Step 3)

Step 2 provides us with predicates $\overline{\mathsf{G}}$ and $\underline{\mathsf{G}}$ satisfying

$$\|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}} \quad = \quad \|\phi\|_{\mathcal{V}} \quad = \quad \|\phi_{\overline{\mathsf{G}}}\|_{\mathcal{V}} \; .$$

We start from the *lhs* and observe that

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \sqcup_{\overline{\sigma}} \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}}^{\overline{\sigma}} \; ,$$

where on the right we omit the now-ignored $\underline{\sigma}$ argument — because the $\sqcap_{\underline{\sigma}}$ can select exactly the predicates provided by $\underline{\mathsf{G}}$ simply by making an appropriate choice of $\underline{\sigma}$.

$\leq$

We then eliminate the explicit strategies altogether by observing that

$$\sqcup_{\overline{\sigma}} \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}}^{\overline{\sigma}} \quad \leq \quad \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}} \quad = \quad \|\phi\|_{\mathcal{V}} ,$$

because the simpler $\|\;\|$-style semantics on the right interprets $\sqcup$ as maximum, which cannot be less than the result of appealing to some strategy function $\overline{\sigma}$, and the equality we have by assumption.

Proof of equivalence (Step 3)

For all $qM\mu$ formulae ϕ, valuations $\mathcal{V}$ and strategies $\underline{\sigma}, \overline{\sigma}$, we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \|\phi\|_{\mathcal{V}} \quad \leq \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; .$$

St

We start from the *lhs* and observe that

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \sqcup_{\overline{\sigma}} \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}}^{\overline{\sigma}} \; ,$$

where on the right we omit the now-ignored $\underline{\sigma}$ argument — because the $\sqcap_{\underline{\sigma}}$ can select exactly the predicates provided by $\underline{\mathsf{G}}$ simply by making an appropriate choice of $\underline{\sigma}$.

$\leq$

We then eliminate the explicit strategies altogether by observing that

$$\sqcup_{\overline{\sigma}} \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}}^{\overline{\sigma}} \quad \leq \quad \|\phi_{\underline{\mathsf{G}}}\|_{\mathcal{V}} \quad = \quad \|\phi\|_{\mathcal{V}} ,$$

because the simpler $\|\;\|$-style semantics on the right interprets $\sqcup$ as maximum, which cannot be less than the result of appealing to some strategy function $\overline{\sigma}$, and the equality we have by assumption.

Proof of equivalence (conclusion)

For all $qM\mu$ formulae ϕ, valuations $\mathcal{V}$ and strategies $\underline{\sigma}, \overline{\sigma}$, we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \|\phi\|_{\mathcal{V}} \quad \leq \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; .$$

Step 3

But trivially from monotonicity we have

$$\sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; ,$$

and so in fact we have shown equality all the way through: we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad = \quad \|\phi\|_{\mathcal{V}} \quad = \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; .$$

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

We have this from Step 1.

Proof of equivalence (concluded)

For all $qM\mu$ formulae ϕ, valuations $\mathcal{V}$ and strategies $\underline{\sigma}, \overline{\sigma}$, we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \|\phi\|_{\mathcal{V}} \quad \leq \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; .$$

But trivially from monotonicity we have

$$\sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; ,$$

and so in fact we have shown equality all the way through: we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad = \quad \|\phi\|_{\mathcal{V}} \quad = \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \; .$$

$=$

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

$=$

$$\sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

Proof of equivalence (concluded)

For all $qM\mu$ formulae ϕ, valuations $\mathcal{V}$ and strategies $\underline{\sigma}, \overline{\sigma}$, we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \|\phi\|_{\mathcal{V}} \quad \leq \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \ .$$

But trivially from monotonicity we have

$$\sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \quad \leq \quad \sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \ ,$$

and so in fact we have shown equality all the way through: we have

$$\sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \ = \ \|\phi\|_{\mathcal{V}} \ = \ \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \|\phi\|_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}} \ .$$

$$= \quad \sqcap_{\underline{\sigma}} \sqcup_{\overline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

Q.E.D.

$$= \quad \sqcup_{\overline{\sigma}} \sqcap_{\underline{\sigma}} \int_{[\![\phi]\!]_{\mathcal{V}}^{\underline{\sigma},\overline{\sigma}}.s} Val$$

Example: Investing in the futures market

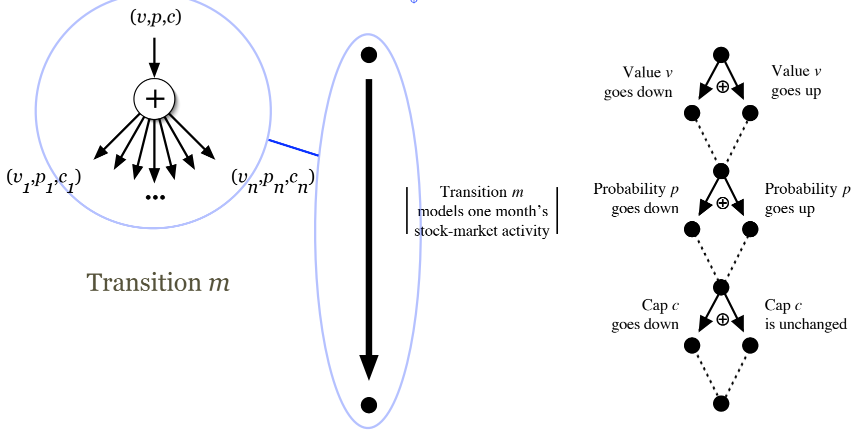
An investor has been given the right to make an investment in “futures”, a fixed number of shares in a specific company that he can *reserve* on the first day of any month he chooses. Exactly *one month later*, the shares will be delivered and will collectively have a *market value* on that day.

His problem is to decide when to make his reservation so that the subsequent market value is *maximised*.

- v shares’ current value
- p probability the value will increase
- c maximum value they can attain, the “cap”

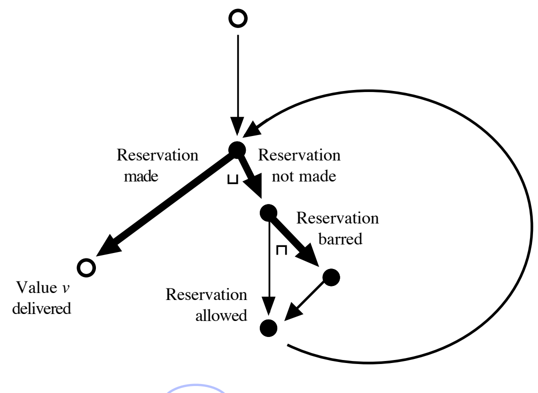
Investing in the futures market

$$\mathcal{R}.S \quad \hat{=} \quad S \to \overline{S_{\$}}$$



Transition m

Investing in the futures market



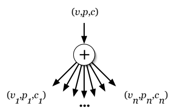
Thick arrows represent transition m from above.

Investing in the futures market

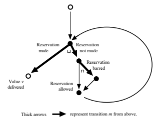
1. The market value v of the shares is a whole number of dollars between \$0 and \$10 inclusive; it has a probability p of going up by \$1 in any month, and $1-p$ of going down by \$1 — but it remains within those bounds. The probability p represents short-term market uncertainty.
2. Probability p itself varies month-by-month in steps of 0.1 between zero and one: when v is less than \$5 the probability that p will rise is 2/3; when v is more than \$5 the probability of p's falling is 2/3; and when v is \$5 exactly the probability is 1/2 of going either way. The movement of p represents investors' knowledge of long-term "cyclic second-order" trends.
3. There is a cap c on the value of v, initially \$10, which has probability 1/2 of falling by \$1 in any month; otherwise it remains where it is. (This modifies Item 1 above.) The "falling cap" models the fact that the company is in a slow decline.
4. If in a given month the investor does not reserve, then at the very next month he might find he is temporarily *barred* from doing so. But he cannot be barred two months consecutively.
5. If he *never* reserves, then he never sells and his return is thus zero.

Investing in the futures market

The semantics for this program fragment is given by *pGCL*, because (deliberately) the probabilistic transitions of *qMμ* have been made to use the *same semantic domain*.



$$
\begin{array}{lll}
\text{month} & \hat{=} & v := (v+1) \sqcap c \ \ {}_{p}\!\oplus\ \ (v-1) \sqcup 0; \\[6pt]
 & & \textbf{if}\ \ \ v<5\ \ \textbf{then}\ \ p := (p{+}0.1) \sqcap 1\ \ {}_{2/3}\!\oplus\ \ (p{-}0.1) \sqcup 0 \\
 & & \textbf{elsif}\ \ v>5\ \ \textbf{then}\ \ p := (p{-}0.1) \sqcup 0\ \ {}_{2/3}\!\oplus\ \ (p{+}0.1) \sqcap 1 \\
 & & \textbf{else}\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ p := (p{-}0.1) \sqcup 0\ \ {}_{1/2}\!\oplus\ \ (p{+}0.1) \sqcap 1 \\
 & & \textbf{fi}; \\[6pt]
 & & c := (c{-}1) \sqcup 0\ \ {}_{1/2}\!\oplus\ \ c
\end{array}
$$



$$
Game\ \ \hat{=}\ \ (\mu X \cdot \{\text{month}\}(v)\ \sqcup\ \{\text{month}\}(X \sqcap \{\text{month}\}X))
$$

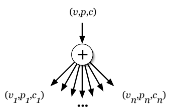
Investing in the futures market

In *Mathematica*

Futures game.nb 2

```
monthPoint[  exp_,      v_, p_, c_   ] := (
        p       *  pUpProbs[v]           *  (1-cFalls)
                *  exp[[makeState[       upV[v,c], upP[p], c                 ]]]

    +   p       *  pUpProbs[v]           *  cFalls
                *  exp[[makeState[       upV[v,c], upP[p], downC[c]     ]]]

    +   p       *  (1-pUpProbs[v])       *  (1-cFalls)
                *  exp[[makeState[       upV[v,c], downP[p], c               ]]]

    +   p       *  (1-pUpProbs[v])       *  cFalls
                *  exp[[makeState[       upV[v,c], downP[p], downC[c] ]]]

    +   (1-p)   *  pUpProbs[v]           *  (1-cFalls)
                *  exp[[makeState[       downV[v], upP[p], c                 ]]]

    +   (1-p)   *  pUpProbs[v]           *  cFalls
                *  exp[[makeState[       downV[v], upP[p], downC[c]          ]]]

    +   (1-p)   *  (1-pUpProbs[v])       *  (1-cFalls)
                *  exp[[makeState[       downV[v], downP[p], c               ]]]

    +   (1-p)   *  (1-pUpProbs[v])       *  cFalls
                *  exp[[makeState[       downV[v], downP[p], downC[c]        ]]]
   )
```

```
monthExp[exp_] := Table[monthPoint[exp, getV[s], getP[s], getC[s]], {s, 1, numStates}]
```



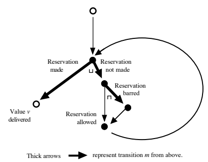
Investing in the futures market

In *Mathematica*

Futures game.nb 3

Constant expectations, and the example iterator functions *f0,f1* etc:

- function *f0* applies the demonic choice late;
- function *f1* applies the demonic choice early;
- function *f2* applies a fixed investor strategy;
- function *f3* calculates probability;
- function *f4* removes the *barring* feature;
- function *f5* calculates probability using a strategy;
- function *f6* makes barring probability 0.5.

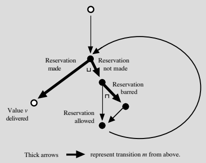



```
vExp = Table[getV[s], {s, 1, numStates}]
constExp[x_] := Table[x, {s, 1, numStates}]
zeroExp = constExp[0]
strategy1[s_] := getC[s] ≤ (getV[s] + 1)
strategy2[s_] := (getV[s] ≥ 5) && (getP[s] ≥ 0.5)
condExp[b_, exp1_, exp2_] := Table[If[ b[s], exp1[[s]], exp2[[s]]], {s, 1, numStates}]
probExp[p_, exp1_, exp2_] := Table[p∗exp1[[s]] + (1 − p)∗exp2[[s]], {s, 1, numStates}]
mvExp = monthExp[vExp]
vAtLeast[v_] := Table[If[getV[s] ≥ v, 1.0, 0.0], {s, 1, numStates}]
mv6Exp = monthExp[vAtLeast[6]]
f0[exp_] := maxExp [mvExp, monthExp[ minExp[exp, monthExp[exp]] ] ]
f1[exp_] := maxExp [mvExp, minExp[monthExp[exp], monthExp[monthExp[exp]] ] ]
f2[exp_] := condExp[strategy1, mvExp, monthExp[ minExp[exp, monthExp[exp]] ] ]
f3[exp_] := maxExp [mv6Exp, monthExp[ minExp[exp, monthExp[exp]] ] ]
f4[exp_] := maxExp [mvExp, monthExp[exp]]
f5[exp_] := condExp[strategy2, mv6Exp, monthExp[ minExp[exp, monthExp[exp]] ] ]
f6[exp_] := maxExp [mvExp, monthExp[ probExp[0.5, exp, monthExp[exp]] ] ]
```

Investing in the futures market

In *Mathematica*

$$Game \;\; \hat{=} \;\; (\mu X \cdot \{\mathsf{month}\}(v) \sqcup \{\mathsf{month}\}(X \sqcap \{\mathsf{month}\}X))$$




```
vExp = Table[getV[s], {s, 1, numStates}]
constExp[x_] := Table[x, {s, 1, numStates}]
zeroExp = constExp[0]
strategy1[s_] := getC[s] ≤ (getV[s] + 1)
strategy2[s_] := (getV[s] ≥ 5) && (getP[s] ≥ 0.5)
condExp[b_, exp1_, exp2_] := Table[If[b[s], exp1[[s]], exp2[[s]]], {s, 1, numStates}]
probExp[p_, exp1_, exp2_] := Table[p∗exp1[[s]] + (1 − p)∗exp2[[s]], {s, 1, numStat
mvExp = monthExp[vExp]
vAtLeast[v_] := Table[If[getV[s] ≥ v, 1.0, 0.0], {s, 1, numStates}]
mv6Exp = monthExp[vAtLeast[6]]
f0[exp_] := maxExp [mvExp, monthExp[ minExp[exp, monthExp[exp]] ] ]
f1[exp_] := maxExp [mvExp, minExp[monthExp[exp], monthExp[monthExp[exp]] ] ]
f2[exp_] := condExp[strategy1, mvExp, monthExp[ minExp[exp, monthExp[exp]] ] ]
f3[exp_] := maxExp [mv6Exp, monthExp[ minExp[exp, monthExp[exp]] ] ]
```


Investing in the futures market

In *Mathematica*

Fixed-point applied to all initial states $0 \leqslant v \leqslant maxV$,
with c at the maximum
and initial share increase probability shown.

```
initialIncreaseProb = 0.5
```

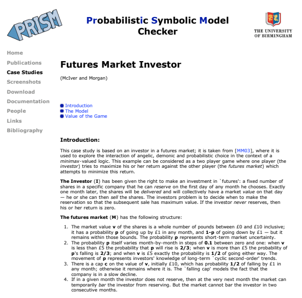
```
fp0 = FixedPoint[f0, zeroExp]
```

```
Do[Print[
    "Initial share value",
    PaddedForm[v, 2],
    " gives highest expected maturity value",
    PaddedForm[fp0[[makeState[v, initialIncreaseProb, maxV]]], {8, 6}],
    "."
  ], {v, 0, maxV}]
```

```
Initial share value  0 gives highest expected maturity value  4.156955.
Initial share value  1 gives highest expected maturity value  4.295363.
Initial share value  2 gives highest expected maturity value  4.553057.
Initial share value  3 gives highest expected maturity value  4.877645.
Initial share value  4 gives highest expected maturity value  5.235896.
Initial share value  5 gives highest expected maturity value  5.523376.
Initial share value  6 gives highest expected maturity value  6.000000.
Initial share value  7 gives highest expected maturity value  7.000000.
Initial share value  8 gives highest expected maturity value  8.000000.
Initial share value  9 gives highest expected maturity value  9.000000.
Initial share value 10 gives highest expected maturity value  9.500000.
```

Investing in the futures market

In *PRISM*



Using the game interpretation, we can generate a pr
above by duplicating nodes with multiple incoming a
minimising or maximising choices as they are encour
the investo ... to whether to inve
whether to ... choose between t
be differen ... their respective d
strategy.

For example, the investor **I**'s strategy (maximising, I
might be: *wait until the share value **v** (rising) meets*
for **v** to rise is a good idea, but when it has met the
further. We call this strategy **I**'s `seat-of-the-pants'

On the other hand, **M**'s strategy (minimising, the inv
*possible, whenever the shares' probability **p** of rising*

As is usual in game theory, when the actual strategi
the *value* of the game to be the the *minimax* over al
but it is well-defined only when it is the same as the
[MM03] the game's value is indeed well-defined.

We have used both Mathematica and PRISM in conju
scripts are available online from here and further de

For example, if **p** is initially **0.5** and the cap **c** is 10,
investor is given in the table below. In the table we
follows the `seat-of-the-pants' strategy given above

```
Initial share value  0 gives highest expected maturity value  4.156955.
Initial share value  1 gives highest expected maturity value  4.295363.
Initial share value  2 gives highest expected maturity value  4.553057.
Initial share value  3 gives highest expected maturity value  4.877645.
Initial share value  4 gives highest expected maturity value  5.235896.
Initial share value  5 gives highest expected maturity value  5.523376.
Initial share value  6 gives highest expected maturity value  6.000000.
Initial share value  7 gives highest expected maturity value  7.000000.
Initial share value  8 gives highest expected maturity value  8.000000.
Initial share value  9 gives highest expected maturity value  9.000000.
Initial share value 10 gives highest expected maturity value  9.500000.
```

initial share value:	expected sale optimal strate
0	4.16
1	4.30
2	4.55
3	4.88
4	5.24
5	5.52
6	6.00
7	7.00
8	8.00
9	9.00
10	9.50

Investing in the futures market

Probabilistic Symbolic Model Checker

THE UNIVERSITY OF BIRMINGHAM

Home
Publications
Case Studies
Screenshots
Download
Documentation
People
Links
Bibliography

Futures Market Investor

(McIver and Morgan)

- Introduction
- The Model
- Value of the Game

Introduction:

This case study is based on an investor in a futures market; it is taken from [MM03], where it is used to explore the interaction of angelic, demonic and probabilistic choice in the context of a *minimax*-valued logic. This example can be considered as a two player game where one player (the *investor*) tries to maximize his or her return against the other player (the *futures market*) which attempts to minimize this return.

The Investor (**I**) has been given the right to make an investment in `futures': a fixed number of shares in a specific company that he can *reserve* on the first day of any month he chooses. Exactly one month later, the shares will be *delivered* and will collectively have a market *value* on that day — he or she can then *sell* the shares. The investors problem is to decide when to make the reservation so that the subsequent sale has maximum value. If the investor *never* reserves, then his or her return is zero.

www.cs.bham.ac.uk/~dxp/prism/casestudies/investor.html

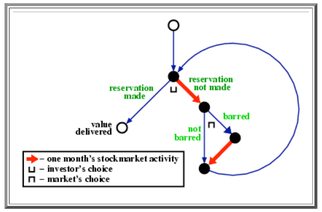
Investing in the futures market

The Model

The situation of the game can be summed up as follows:

- During each month there are three probabilistic transitions that occur: the share value **v** rises or falls, according to **p**; probability **p** itself rises or falls, according to long-term trends; and the capped value **c** of the stock either falls or stays the same. The compounded effect of these choices determine a transition represting one month's stockmarket activity.
- At the beginning of each month, the investor makes a maximising choice of whether to reserve; but, if he does not, then
- At the beginning of the next month, the market makes a minimising choice of whether to bar the investor or not.

Graphically, the behaviour of the model is given by the transition system below.



Investing in the futures market

Modelling the system in PRISM

Below we give the PRISM code for this model. The model has 6,688 states and in PRISM it takes 0.169 seconds to construct this model.

```
// EXAMPLE: INVESTING IN THE FUTURES MARKET
// (McIver and Morgan 03)

nondeterministic

// module use to synchronize transitions
module month

    m : [0..1];

    [invest] (m=0) -> (m'=1); // transitions made at the start of the month synchronize on 'invest'
    [month] (m=1) -> (m'=0); // transitions made during the month synchronize on 'month'
    [done] (m=0) -> (m'=0); // once investor has cashed in shares nothing changes

endmodule

// the investor
module investor

    i : [0..1]; // i=0 no reservation and i=1 made reservation

    [invest] (i=0) -> (i'=0); // do nothing
    [invest] (i=0) -> (i'=1); // make reservation
    [invest] (i=1) & (b=1) -> (i'=0); // barred previous month: try again and do nothing
    [invest] (i=1) & (b=1) -> (i'=1); // barred previous month: make reservation
    [done] (i=1) & (b=0) -> (i'=1); // cash in shares (not barred)

endmodule

// bar on the investor
```


Investing in the futures market

Value of the Game

Using the game interpretation, we can generate a probabilistic tree from the transition system above by duplicating nodes with multiple incoming arcs, `unfolding' back-loops, and making minimising or maximising choices as they are encountered. In this example, at each unfolding both the investor **I** (making choices as to whether to invest) and the stock market **M** (making choices whether to impose a bar) need to choose between two ongoing branches — and their choice could be different each time they revisit their respective decision points. Each of **I** and **M** will be using a *strategy*.

For example, the investor **I**'s strategy (maximising, he or she hopes) for dealing with the falling cap might be: *wait until the share value **v** (rising) meets the cap **c** (falling), and then reserve.* Waiting for **v** to rise is a good idea, but when it has met the cap **c** there is clearly no point in waiting further. We call this strategy **I**'s `seat-of-the-pants' strategy.

On the other hand, **M**'s strategy (minimising, the investor fears) might be: *bar the investor, if possible, whenever the shares' probability **p** of rising exceeds 1/2*.

As is usual in game theory, when the actual strategies of the two players are unknown, we define the *value* of the game to be the the *minimax* over all strategy sequences of the expected payoff — but it is well-defined only when it is the same as the *maximin*. From the results presented in [MM03] the game's value is indeed well-defined.

We have used both Mathematica and PRISM in conjuction with Matlab for this calculation. The scripts are available online from here and further details are available in [MM03].

For example, if **p** is initially **0.5** and the cap **c** is 10, then the optimal expected sale-value for the investor is given in the table below. In the table we also include the results when the investor follows the `seat-of-the-pants' strategy given above.

initial share value:		**expected sale price:**	
		optimal strategy	**`seat-of-the-pants' strategy**
0		4.16	3.68
1		4.30	3.79
2		4.55	3.97
3		4.88	4.17
4		5.24	4.29
5		5.52	4.17
6		6.00	4.16
7		7.00	4.65
8		8.00	5.61
9		9.00	6.78
10		9.50	9.50

Investing in the futures market

For example, the investor **I**'s strategy (maximising, he or she hopes) for dealing with the falling cap might be: *wait until the share value **v** (rising) meets the cap **c** (falling), and then reserve.* Waiting for **v** to rise is a good idea, but when it has met the cap **c** there is clearly no point in waiting further. We call this strategy **I**'s `seat-of-the-pants' strategy.

For example, if **p** is initially **0.5** and the cap **c** is 10, then the optimal expected sale-value for the investor is given in the table below. In the table we also include the results when the investor follows the `seat-of-the-pants' strategy given above.

initial share value:		**expected sale price:**	
		optimal strategy	**`seat-of-the-pants' strategy**
0		4.16	3.68
1		4.30	3.79
2		4.55	3.97
3		4.88	4.17
4		5.24	4.29
5		5.52	4.17
6		6.00	4.16
7		7.00	4.65
8		8.00	5.61
9		9.00	6.78
10		9.50	9.50

Exercises

Ex. 1: *Work through the creation of the tree for the formula given, and show that it is equivalent to the one depicted.*