From Number Guessing Games

to

Security Protocol Analysis

and

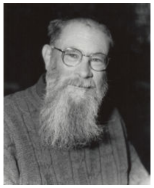
Back

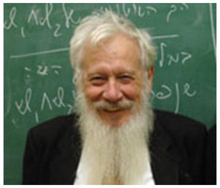
Jan van Eijck
CWI & ILLC, Amsterdam

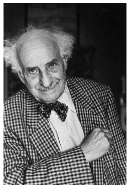
R60 Jubilee Event, Tilburg, December 16, 2013

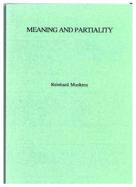
Abstract

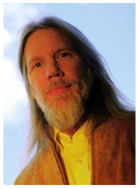
The talk will explain what goes on in number guessing games, and present a new formal representation (in terms of Kripke semantics) of information update in such games. This can be used in real life settings: one application is the analysis of cryptographic security protocols. I will look at a protocol for secret key distribution over an insecure network, and show how this is analysed with Kripke semantics. Finally, I address the question whether there are lessons to be learned for natural language analysis.







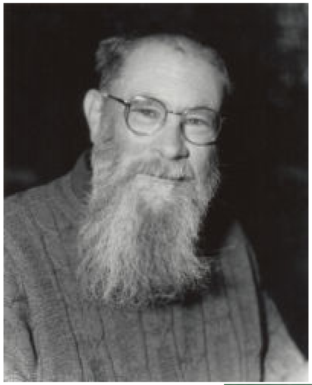

MEANING AND PARTIALITY

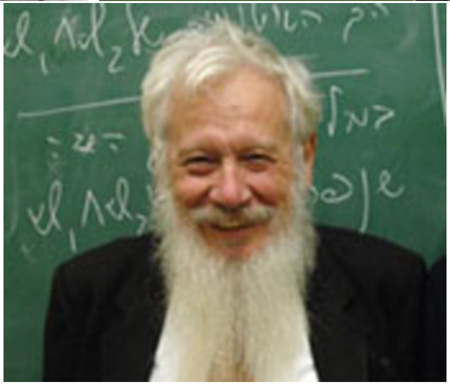
Reinhard Muskens


Language and Communication

No essential difference between formal and natural languages for communication.

What does it mean to know a proposition?





How to Model Ignorance?

$$\begin{tikzcd} w : H \arrow[r, dash] & w' : T \end{tikzcd}$$

Probability, Betting, Ignorance

Probability, Betting, Ignorance

Christiaan Huygens, 1629–1695

Van Rekeningh in Spelen van Geluck

1660
[Huy60]

beroemden Heer CHRISTIANVS HVGENIVS aengaende 't reeckenen in Spelen van Geluck uytgevonden ende my in ſchrift van hem mede gedeelt is, alhier met deſſelfs brief, in plaets van 'tgeene my overig was, by-voegde. Welck ſijn Tractaet ick dan UE. des te aengenamer acht te ſullen weſen, als 'tgeen daer in verhandelt wordt te ſubtijlder en ongemeender ſal bevonden worden; inſonderheyt door dien hy tot deſſelfs vinding deſelve *Analyſis*, welckers fondamenten hy certijts van my geleert heeft, als ick, gebruyckt; en alſoo de bevlytigers van die de weg baent om diergelijcke Voorſtellen te ontbinden. Waer in, ſoo ick nevens mijnen anderen arbeyt aen U, Beminde Leſer, in deſe ſoort van Studie genoegſame ſtof van oeffening gegeven hebbe; ſoo ſult ghy daer uyt (gelijck ick hoop) mijne bereytwilligheyt t'uwaerts konnen afnemen, en dienvolgende oock mijnen arbeyt, t'uwen en der Studien beſten aengenomen, ten goeden duyden. Vaert wel.

Aen

Aen mijn Heer

FRANCISCUS van SCHOOTEN.

Mijn Heer,

NAer dien ick weet dat V E., de loffelijcke vruchten van ſijn vernuft ende arbeyt in 't licht gevende, onder anderen dit ooghmerck heeft: namentlijck, om door de verſcheydenheyt der verhandelde ſtoffen te betoonen hoe wijt onſe uytnemende Konſt van Algebra *ſich uytſtreckt; ſoo en twijffele ick oock niet, of het geene ick van de Rekeningh in Spelen van geluck beſchreven heb, ſal tot V E. opſet niet ondienſtig zijn. Want ſoo veel te ſwaerder als het ſchijnt, door reden te konnen bepalen het geene onſeker is ende het geval onderworpen, ſoo veel te meer verwonderinghs waerdigh ſal die weetenſchap ſchijnen, waer door ſulcx kan werden te weege gebracht. Dewijl ick dan op V E. verſoeck ende aen-maeninghe, deſe Rekening eerſt heb beginnen by geſchrift te ſtellen, ende V E. deſelve waerdigh acht om te gelijck met ſyne diepſinnige vonden in 't licht te komen; ſoo en ſal ick niet alleen het ſelve geerne toeſtaen, maer oock tot mijn voordeel duyden, dat die op deſe maniere te voorſchijn werde gebracht. Want of ſommige mochten dencken dat ick ontrent geringhe dingen en van weynigh gewichte mijn moeyte beſteedt hadde, ſoo en ſullen ſy nochtans niet t'eenemael voor onnut ende onpryſſelijck houden, het geene V E. in dier voegen als voor het ſyne is aen-nemende, en niet ſonder arbeyt uyt onſe ſpraeck in de Latijnſche heeft overgeſet. Alhoewel ick wil gelooven, ſo iemandt deſe dinghen wat naerder begint in te ſien, dat hy haeſt ſal bevinden, geen enckel ſpel te zijn het geene hier wert verhandelt, maer datter de beginſelen en gronden geleyt werden van een ſeer aerdige en diepe ſpeculatie. Soo ſullen oock, meyne ick, de Voorſtellen die in deſe Materie voorvallen, geenſins lichter als die van* Diophantus *geacht werden, doch wel vermaeckelijcker miſſchien, door dienſt tets meer inhouden als bloote eygenſchappen der getallen. Voorts is te weten,*

dat

Huygens on the Foundations of Probability

“Ick neeme tot beyder fondament, dat in het speelen de kansse, die yemant ergens toe heeft, even soo veel weerdt is als het geen, het welck hebbende hy weder tot deselfde kansse kan geraecken met rechtmatigh spel, dat is, daer in niemandt verlies geboden werdt. By exempel. So yemandt sonder mijn weeten in d’eene handt 3 schellingen verbergt, en in d’ander 7 schellingen, ende my te kiesen geeft welck van beyde ick begeere te hebben, ick segge dit my even soo veel weerdt te zijn, als of ick 5 schellingen seecker hadde. Om dat, als ik 5 schellingen hebbe, ick wederom daer toe kan geraecken, dat ick gelijcke kans sal hebben, om 3 of 7 schellingen te krijgen, en dat met rechtmatigh spel: gelijck hier naer sal betoont werden.”

Translation

“I take as the foundation of both [calculating what non-finished hazard games are worth, and calculating winning chances in such games] that in playing the chance that someone has in some matter, is worth just as much as the amount that, if he possesses it, will give him the same chances in a fair game, that is a game where no loss is offered to anyone. For instance. Suppose someone without my knowing hides in one hand 3 shillings, and in the other 7 shillings, and he offers me the choice between the two hands. Then I would say that this offer is worth the same as having 5 shillings for sure. Because, if I have 5 shillings, I can wager them in such manner that I have equal chances of getting 3 or 7 shillings, and that in a fair game, as will be explained hereafter.”

Huygens’ First Proposition

“Als ick gelijcke kans hebbe om a of b te hebben, dit is my so veel weerdt als $(a + b)/2$”

“If I have equal chances of getting a and b, then to me this is worth $(a + b)/2$.”

Next, Huygens gives an explanation in terms of a game, and uses this explanation to prove the proposition.

Explanation and Proof in Terms of a Game

“Want (a+b)/2 hebbende, soo kan ick dat tegen een ander waegen die mede (a+b)/2 sal insetten, ende bedingen dat die het spel wint, den anderen sal a geven. Waer door ick gelijcke kans sal bekomen om a te hebben, te weeten, als ick verlies, of b als ick win; want alsdan soo treck ick a+b dat in-geset is, ende geef hem daer van a.”

Tr:

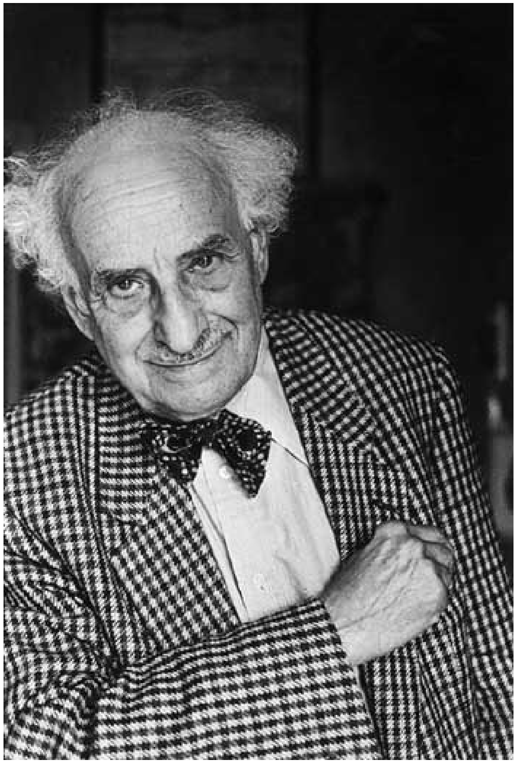
“For having $(a+b)/2$, I can wager that against someone else who also stakes $(a+b)/2$, with the stipulation that the one who wins the game will give the other a. This will give me equal chance of getting a, namely if I lose, or b if I win; because in that case I get the stake of $a+b$, and from this I give him a.”

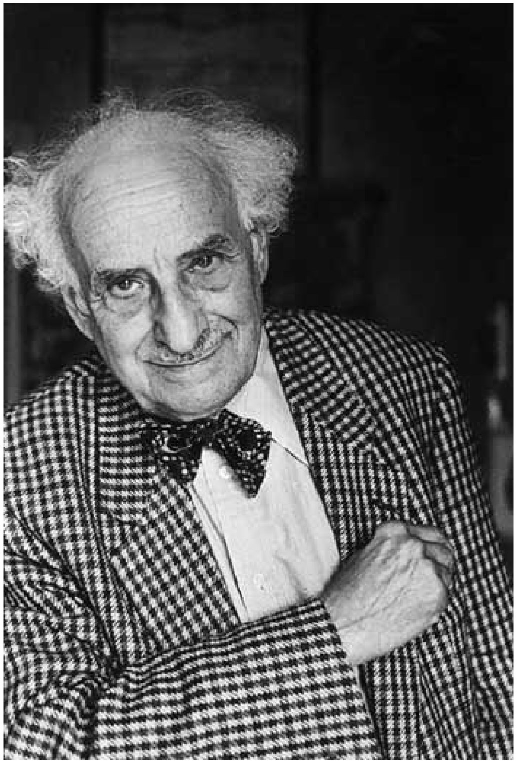
Example in Numbers

“[. . .] In getaelen. Indien ick gelijcke kans heb om 3 te hebben of 7, soo is door dit Voorstel mijn kansse 5 weerdt; ende het is seecker dat ick 5 hebbende weder tot de selfde kansse kan geraecken. Want speelende om de selve tegen een ander die daer 5 tegen set, met beding dat de geene die wint den anderen 3 sal geven; soo is dit rechtmaetig spel, ende het blijckt dat ick gelijcke kans hebbe om 3 te hebben, te weeten, als ick verlies, of 7 indien ick win; want alsdan treck ick 10, daer van ick hem 3 geef.”

Example in Numbers: Translation

“[. . .] In numbers. If I have equal chances to have 3 or 7, then by my Proposal this chance is worth 5; and it is sure that if I have 5, I will get to the same chance. Because putting 5 at stake against someone who stakes 5 against it, with condition that the one who wins will give the other 3, one has a fair game, and it becomes clear that I have equal chance of getting 3, namely, if I lose, or 7 if I win; because if I win I draw 10, of which I give 3 to him.”




Hans Freudenthal, 1905–1990

Comment of Hans Freudenthal

“This is not a trivial transformation. The original situation for which the expectation has to be calculated is that of an individual in a game-like situation, yet with no adversary identified (say a lottery). It is reconstructed as an n persons’ game (if n is the number of chances proposed) with equal stakes and the appropriate stipulations, which by virtue of their mutuality are fair. The value of the stakes under consideration is just the expectation.

“Equal chance,” as used by Huygens, does not beg the question either. In Huygens’ model, where the player is given the free choice to take one hand or the other, or, more generally, to decide himself which one of a number of possibilities he chooses, equal chance is validly defined by the complete ignorance and the free will of the player.”

[Fre80]

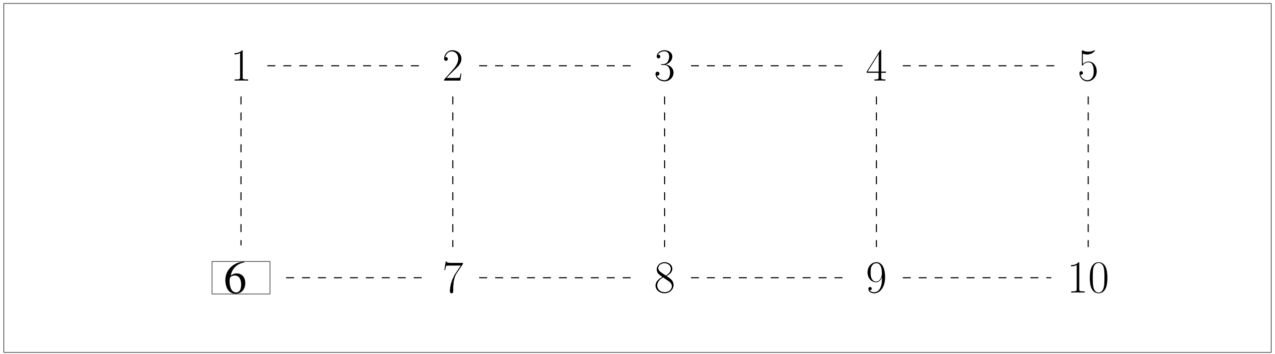
What does it mean to know a number?

Jan, Gaia and Rosa Play the Number Guessing Game

Number Guessing Game, Naive Version

- Jan says: “I have a number in mind, in the range from one to ten. You may take turns guessing. Whoever guesses the number first gets the toy. It is Gaia’s turn to start, for last time we played this game Rosa started the guessing.”
- Gaia and Rosa agree …
- After a number of rounds, Jan announces: “Rosa, you have won.”

Number Guessing: Naive Representation



Number Guessing: Naive Representation

$$\begin{tikzcd}
1 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 2 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 3 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 4 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 5 \arrow[d, dash, dashed] \\
\boxed{6} \arrow[r, dash, dashed] & 7 \arrow[r, dash, dashed] & 8 \arrow[r, dash, dashed] & 9 \arrow[r, dash, dashed] & 10
\end{tikzcd}$$

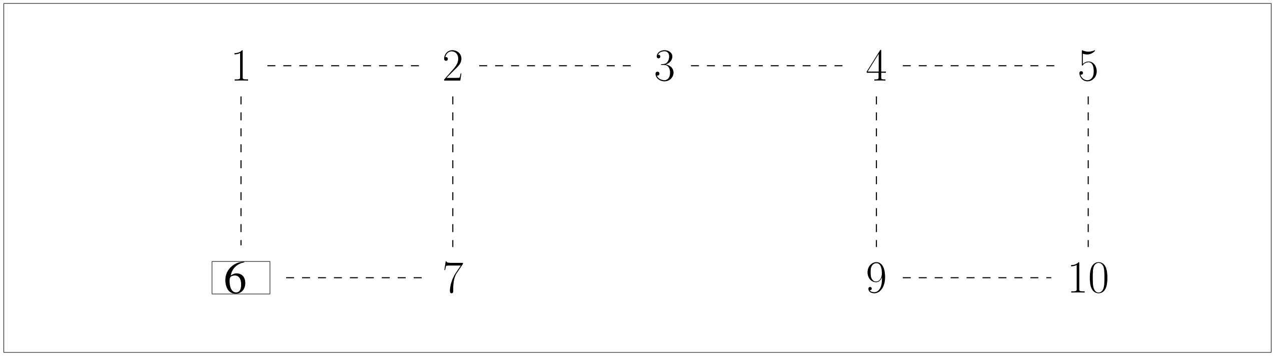
“eight” … “no”

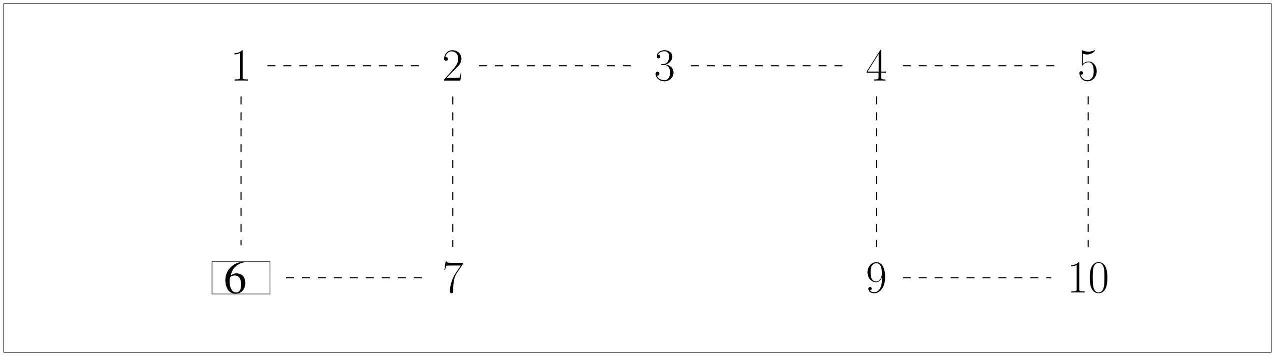
Number Guessing: Naive Representation

$$\begin{tikzcd}
1 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 2 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 3 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 4 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 5 \arrow[d, dash, dashed] \\
\boxed{6} \arrow[r, dash, dashed] & 7 \arrow[r, dash, dashed] & 8 \arrow[r, dash, dashed] & 9 \arrow[r, dash, dashed] & 10
\end{tikzcd}$$

“eight” . . . “no”

$$\begin{tikzcd}
1 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 2 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 3 \arrow[r, dash, dashed] & 4 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 5 \arrow[d, dash, dashed] \\
\boxed{6} \arrow[r, dash, dashed] & 7 & & 9 \arrow[r, dash, dashed] & 10
\end{tikzcd}$$


1
2
3
4
5
6
7
9
10



“five” … “no”

$$\begin{tikzcd}
1 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 2 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 3 \arrow[r, dash, dashed] & 4 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 5 \arrow[d, dash, dashed] \\
\boxed{6} \arrow[r, dash, dashed] & 7 & & 9 \arrow[r, dash, dashed] & 10
\end{tikzcd}$$

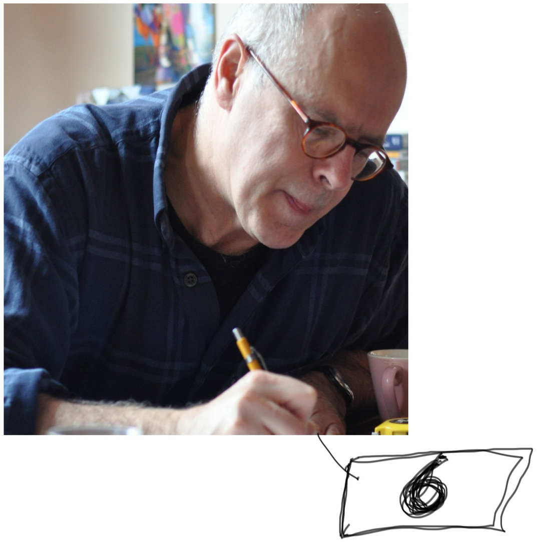
"five" … "no"

$$\begin{tikzcd}
1 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 2 \arrow[r, dash, dashed] \arrow[d, dash, dashed] & 3 \arrow[r, dash, dashed] & 4 \arrow[d, dash, dashed] & \\
\boxed{6} \arrow[r, dash, dashed] & 7 & & 9 \arrow[r, dash, dashed] & 10
\end{tikzcd}$$

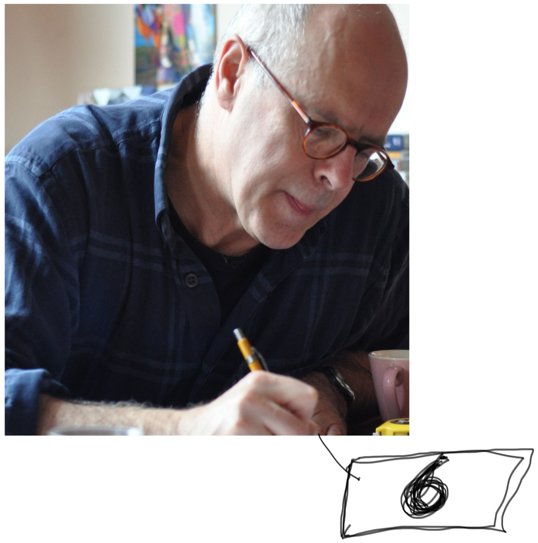
Number Guessing Game, More Sophisticated Version

- Jan says: “I have a number in mind, in the range from one to ten. You may take turns guessing. Whoever guesses the number first gets the toy. It is Gaia’s turn to start, for last time we played this game Rosa started the guessing.”
- Gaia does not agree. “How can we know you are not cheating on us? Please write down the number, so you can show it to us afterwards as a proof.”
- Jan writes a number on a piece of paper, hidden from Gaia and Rosa.
- After a number of rounds, Jan announces: “Rosa, you have won.”
- Jan shows the piece of paper as a proof.

Jan, Gaia and Rosa Play the Number Guessing Game (again)



Jan, Gaia and Rosa Play the Number Guessing Game (again)



register

Registers, Fixing a Number

At the point where either Gaia or Rosa demands that the secret number gets written down, and that Jan shows it as a proof that the procedure was honest, the important notion of a register arises.

The register allows Jan to prove that he knew (had fixed) the number beforehand.

What Does it Mean to Know a Number?

We are working in the context of epistemic logic.

> To know an integer number n is to be able distinguish a world where n is written in some register from all possible worlds where an integer number n' different from n is written in that register.

This can be made more precise, by clearly distinguishing between a register and its contents.

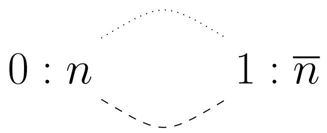
And the representation can be made more concise, by lumping together all situations where the register does not contain the number.

> To know an integer number n is to know the contents $\dot{n}$ of register n, not to know a number is not to have access to register n.

Jan knows a number

Jan knows n, the two children do not know n. We use n for the register, $\dot{n}$ for the value of that register (its contents).

This leads to the following register model.



$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$

0 is a world that has register n, with number $\dot{n} = 6$ stored in it, while 1 differs from 0 in that it also has this register, but with all the possible values in it that differ from $\dot{n}$.

Gaia (dots) and Rosa (dashes) do not have access to the register.

$$\begin{tikzcd}
0 : n \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n}
\end{tikzcd}$$

$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$

$$\begin{tikzcd} 0 : n \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n} \end{tikzcd}$$

$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$

“ten” … “no”

$$\begin{tikzcd}
0 : n \arrow[r, bend left, dotted, no head] \arrow[r, bend right, dashed, no head] & 1 : \overline{n}
\end{tikzcd}$$

$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$

“ten” … “no”

$$\begin{tikzcd}
0 : n \arrow[r, bend left, dotted, no head] \arrow[r, bend right, dashed, no head] & 1 : \overline{n}
\end{tikzcd}$$

$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9\}$$

$$\begin{tikzcd}
0 : n \arrow[r, dotted, no head, bend left] \arrow[r, dashed, no head, bend right] & 1 : \overline{n}
\end{tikzcd}$$

$n \mapsto 6$
$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$

“ten” … “no”

$$\begin{tikzcd}
0 : n \arrow[r, dotted, no head, bend left] \arrow[r, dashed, no head, bend right] & 1 : \overline{n}
\end{tikzcd}$$

$n \mapsto 6$
$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9\}$

“six” … “yes”

$$\begin{tikzcd} 0 : n \arrow[r, dotted, bend left, no head] \arrow[r, dashed, bend right, no head] & 1 : \overline{n} \end{tikzcd}$$

$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$

"ten" … "no"

$$\begin{tikzcd} 0 : n \arrow[r, dotted, bend left, no head] \arrow[r, dashed, bend right, no head] & 1 : \overline{n} \end{tikzcd}$$

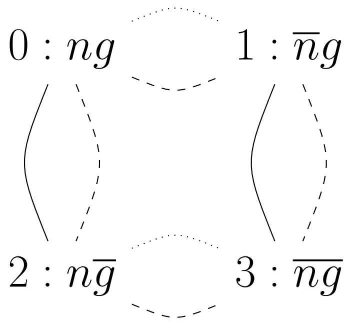
$$n \mapsto 6$$
$$\overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9\}$$

"six" … "yes"

$$0 : n \qquad\qquad n \mapsto 6$$

Gaia Makes a Guess

Use g for the guess of Gaia. Gaia knows her guess, but has not yet revealed it to the others: solid lines for Jan, dotted lines for Gaia, dashed lines for Rosa.



$$n \mapsto 6, \overline{n} \mapsto \{1,2,3,4,5,7,8,9,10\}$$
$$g \mapsto 5, \overline{g} \mapsto \{1,2,3,4,6,7,8,9,10\}$$

Without registers, this blows up to a model with 100 worlds.

k registers, each of m bits, blow up to $(2^m)^k$ possibilities.

Gaia Reveals Her Guess

Gaia reveals the contents of her register: $g = 5$.

Gaia Reveals Her Guess

Gaia reveals the contents of her register: $g = 5$.

$$\begin{tikzcd}
0 : ng \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n}g
\end{tikzcd}$$

$n \mapsto 6, \overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$

$g \mapsto 5$

Gaia Reveals Her Guess

Gaia reveals the contents of her register: $g = 5$.

$$\begin{tikzcd} 0 : ng \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n}g \end{tikzcd}$$

$$n \mapsto 6, \overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$
$$g \mapsto 5$$

Jan states that the guess is wrong: $n \neq g$.

Gaia Reveals Her Guess

Gaia reveals the contents of her register: $g = 5$.

$$\begin{tikzcd}0 : ng \arrow[r, bend left, dotted, no head] \arrow[r, bend right, dashed, no head] & 1 : \overline{n}g\end{tikzcd}$$

$$n \mapsto 6, \overline{n} \mapsto \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$$
$$g \mapsto 5$$

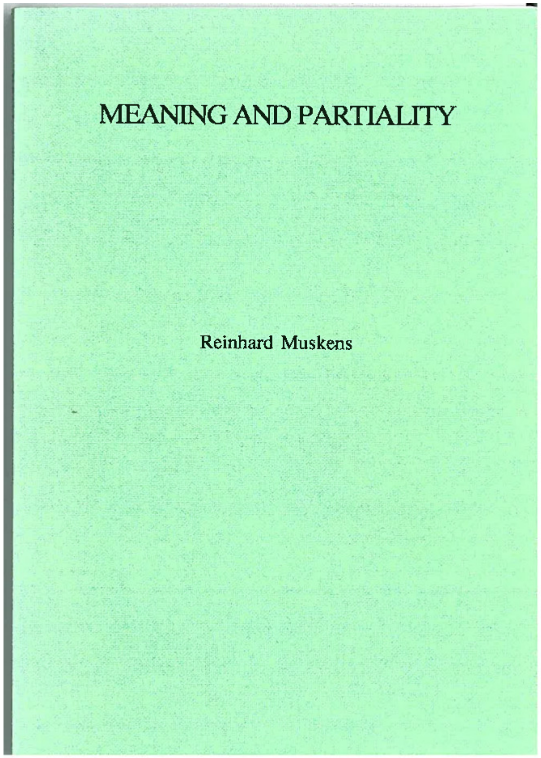
Jan states that the guess is wrong: $n \neq g$.

$$\begin{tikzcd}0 : ng \arrow[r, bend left, dotted, no head] \arrow[r, bend right, dashed, no head] & 1 : \overline{n}g\end{tikzcd}$$

$$n \mapsto 6, \overline{n} \mapsto \{1, 2, 3, 4, 7, 8, 9, 10\}$$
$$g \mapsto 5$$

Names and the Meaning of Equality Statements [Mus89]

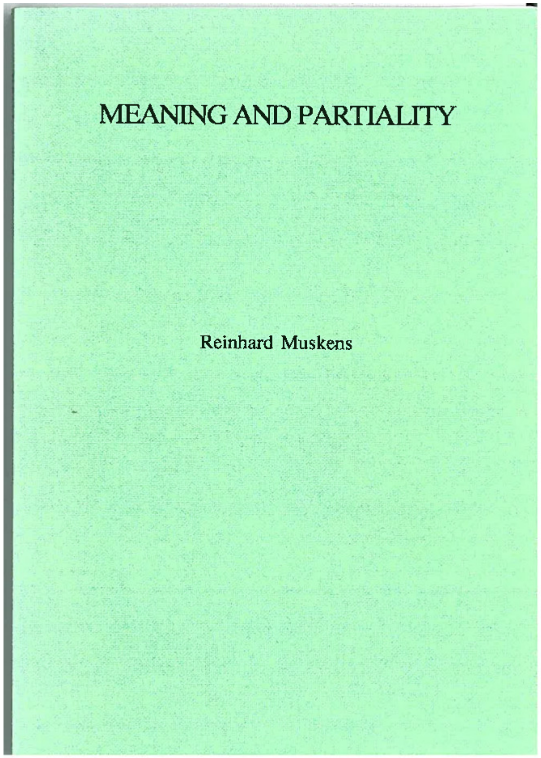
Names and the Meaning of Equality Statements [Mus89]



MEANING AND PARTIALITY

Reinhard Muskens

Names and the Meaning of Equality Statements [Mus89]


MEANING AND PARTIALITY

Reinhard Muskens

Chapter 8: Names.

Registers are like Names for Numbers

- Creating a register and writing a number in it can be viewed as a baptism.
- Register equality statements are like name equality statements.
- Compare $n = g$ with “Hesperus is Phosphorus”.
- Reread Chapter 8 of Reinhard’s PhD Thesis.

Truth of Equality Statements in Register Models

- It seems reasonable to represent the announcement of Gaia's guess as the announcement of the equality statement $g = \dot{g}$.
- Since $\overline{g}$ means that the register does contain a different value from $\dot{g}$, the statement $g = \dot{g}$ is false in both $\overline{g}$ worlds, so these drop out of the picture.
- Jan can respond to Gaia's guess with $n = g$ (this is what "you guessed it" means: your guess equals my number).
- $n = g$ is different from $n = \dot{n}$.

Truth of (In)equality Statements in Register Models

- The difference between $n = g$ and $n = \dot{n}$ can perhaps be seen more clearly in the negative case.
- "You have it wrong" is paraprased as $n \neq g$, not as $n \neq \dot{n}$.
- In a world where g and n are both true, the equality $n = g$ can be checked by checking whether $\dot{g} = \dot{n}$.
- In a world where g is true and n false, the equality $n = g$ means that every value of n except for $\dot{n}$ equals g, which is certainly false (except in a domain with just two numbers, in case $\dot{n}$ and $\dot{g}$ are different).
- In a world where g and n are both false the equality $n = g$ means that every value for g except $\dot{g}$ equals every value of n except $\dot{n}$, which is certainly false.

Register Language for Guessing Games

Let p range over $\mathbf{P}$ and let N range over $\mathbb{Z}$. Let i range over the set of agents I.

$$\varphi \;::=\; \top \mid p \mid p = E \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [!p = E]\varphi \mid [!p \neq E]\varphi$$

$$E \;::=\; p \mid N$$

Update Operations

Revealing positive or negative information about p:

$! \, p = E$

“You guessed it”

$! \, p \neq E$

“Your guess was wrong”

Interpretations of $! \, p = E$ and $! \, p \neq E$ are action models in the sense of [BMS98].

Interpretation in Register Models

- A register assignment function is a function from registers to integers.
- A register assignment function h agrees with a possible w (notation $w \multimap h$) if $q \in V_w$ iff $h(q) = \dot{q}$.
- Examples of (Dis-)Agreement:

$$\begin{tikzcd}
0 : ng \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n}g
\end{tikzcd}$$

$$n \mapsto 6, \overline{n} \mapsto \{1..4, 7..10\}$$
$$g \mapsto 5$$

The function $h = \{n \mapsto 6, g \mapsto 5\}$ agrees with 0, but not with 1.

The function $h' = \{n \mapsto 3, g \mapsto 5\}$ agrees with 1 but not with 0.

Interpretation in Register Models: $\mathcal{M}, w, h \models \varphi$

Let h be a register assignment that agrees with w.

$$\begin{array}{rcl}
\mathcal{M}, w, h \models \top & & \text{always} \\
\mathcal{M}, w, h \models p & \text{iff} & p \in V(w) \\
\mathcal{M}, w, h \models p_1 = p_2 & \text{iff} & h(p_1) = h(p_2) \\
\mathcal{M}, w, h \models p = N & \text{iff} & h(p) = N \\
\mathcal{M}, w, h \models \neg\varphi & \text{iff} & \text{not } \mathcal{M}, w, h \models \varphi \\
\mathcal{M}, w, h \models \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{M}, w, h \models \varphi_1 \text{ and } \mathcal{M}, w, h \models \varphi_2 \\
\mathcal{M}, w, h \models K_i\varphi & \text{iff} & (w, w') \in R_i \text{ and } w' \multimap h' \text{ imply } \mathcal{M}, w', h' \models \varphi \\
\mathcal{M}, w, h \models [!\, p = E]\varphi & \text{iff} & \mathcal{M}, w, h \models p = E \text{ implies } \mathcal{M}^{p=E}, w, h \models \varphi \\
\mathcal{M}, w, h \models [!\, p \neq E]\varphi & \text{iff} & \mathcal{M}, w, h \models p \neq E \text{ implies } \mathcal{M}^{p\neq E}, w, h \models \varphi
\end{array}$$

Truth and Falsity at a World

From truth at a register assignment to truth and falsity at a world:

$$\mathcal{M}, w \models \varphi \text{ iff } \forall h \text{ with } w \multimap h : \mathcal{M}, w, h \models \varphi.$$

$$\mathcal{M}, w \dashv \varphi \text{ iff } \forall h \text{ with } w \multimap h : \mathcal{M}, w, h \models \neg\varphi.$$

Truth Value Gaps

$$\begin{tikzcd}
0 : n \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n}
\end{tikzcd}$$

$$n \mapsto 6$$
$$\overline{n} \mapsto \{1..5, 7..10\}$$

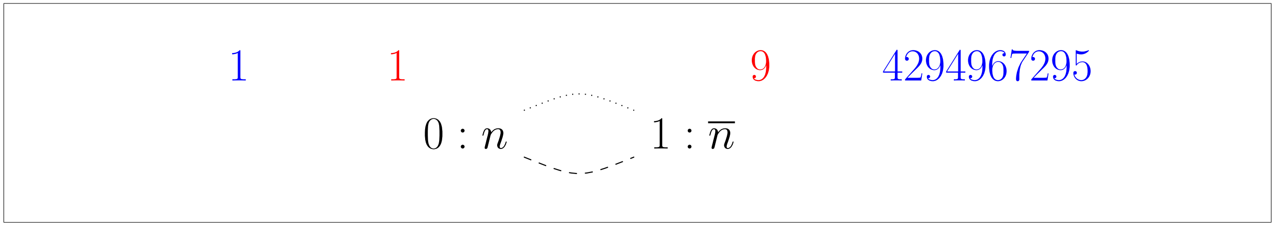
$n = 7$ is false in 0.

$n = 7$ is neither true nor false in 1.

$n \neq 7$ is true in 0.

$n \neq 7$ is neither true nor false in 1.

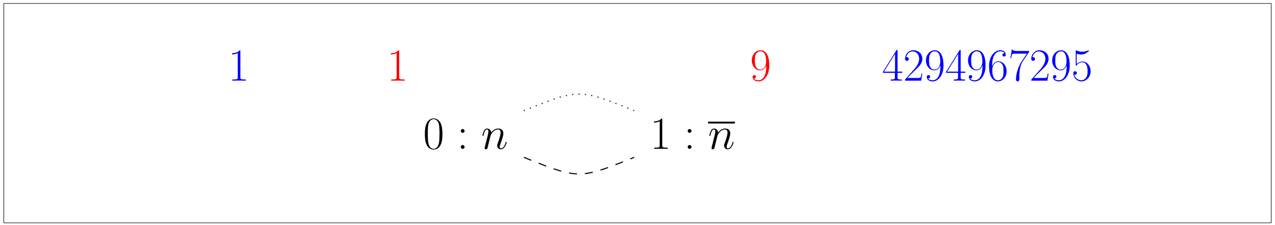
Now Let’s Look at Probabilities



Numbers of agreeing assignments for register size 10.

Numbers of agreeing assignments for register size of 32 bits.

Now Let’s Look at Probabilities



Numbers of agreeing assignments for register size 10.

Numbers of agreeing assignments for register size of 32 bits.

With 64 bit size: $2^{64} - 1 = 18446744073709551615$ agreeing assignments for world 1.

Back to Christiaan Huygens

- Suppose possible guesses run from 1 to 10.
- A fair guessing game would have 10 participants. Each guess corresponds with a ticket. Winning ticket gets the stake.
- Then a ticket is worth $\frac{1}{10}$th of the stake.
- World 0 corresponds to a single ticket, world 1 corresponds to a set of 9 tickets.
- With a register size of 64 bits, a ticket is worth $1/18446744073709551616$ of the stake, and world 0 corresponds to a single ticket, world 1 corresponds to 18446744073709551615 tickets.
- This can be worked out in a logic that combines probability theory with epistemic operators [Hal03, Koo03, ES].

Monte Carlo Checking of (In)Equality Statements

Suppose the register size is large enough, say 64 bits.

Then in a world where n is false, there are

$$2^{64} - 1 = 18446744073709551615$$

possibilities for the value of n.

Thus, in a world where n is false, equality statements $n = X$ will be false for almost all values X.

Equality and inequality statements can be checked by means an approximate (Monte Carlo) model checking algorithm.

Monte Carlo Style Model Checking

$$\mathcal{M}, w \stackrel{\cdot}{\models} \varphi \text{ iff for ``enough'' } h \text{ with } w \multimap h : \mathcal{M}, w, h \models \varphi.$$

Method:

- Randomly generate a list of n agreeing assignments $h_1, \ldots, h_n$ for w.
- Check $\mathcal{M}, w, h_i \models \varphi$ for each h_i in the list.
- The probability that there is disagreement among the outcomes can be made arbitrarily small.

Truth Value Gaps Almost Closed

$$\begin{tikzcd}
1 & \\
0 : n \mapsto 6 \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n} \mapsto \{M = -2^{63}..2^{63}-1\} - \{6\} \\
 & 18446744073709551615
\end{tikzcd}$$

$n = 7$ is false in 0.

$n = 7$ is almost false in 1.

$n \neq 7$ is true in 0.

$n \neq 7$ is almost true in 1.

Article of faith of cryptanalysis: “Brute force attacks on number secrets are impossible.”

Register Creation

Fix a large register size. Let $M..K$ be the bounds of the register (say $M = -2^{63}$, $K = 2^{63} - 1$).

Add to the language:

$$\varphi ::= [p \overset{i}{\leftarrow} N]\varphi$$

$p \overset{i}{\leftarrow} N$ is the command to link p to the integer number N with the link known only to agent i. It is assumed that N fits the register size.

Interpretation: $[\![p \overset{i}{\leftarrow} N]\!]$ as the action model (in the sense of [BMS98]):

$$\begin{tikzcd} \boxed{\text{e}} : p := \top, \dot{p} := N \arrow[r, dash, "I - \{i\}"] & e : p := \bot \end{tikzcd}$$

Update Example

Start with the blissful ignorance unit model:

$$\boxed{0}$$

Update Example

Start with the blissful ignorance unit model:

$$\boxed{0}$$

$n \overset{j}{\leftarrow} 6$

Update Example

Start with the blissful ignorance unit model:

$$0$$

$$n \xleftarrow{j} 6$$

$$\begin{tikzcd} 0 : n \arrow[rr, dash, dotted, bend left] \arrow[rr, dash, dashed, bend right] & & 1 : \overline{n} \end{tikzcd} \qquad n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$$

Update Example

Start with the blissful ignorance unit model:

$$\boxed{\quad 0 \quad}$$

$n \overset{j}{\leftarrow} 6$

$$\begin{tikzcd} 0 : n \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n} \end{tikzcd} \qquad n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$$

$g \overset{g}{\leftarrow} 5$

Update Example

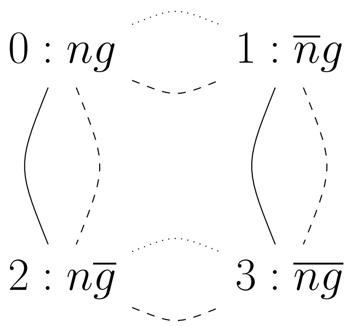
Start with the blissful ignorance unit model:

$$0$$

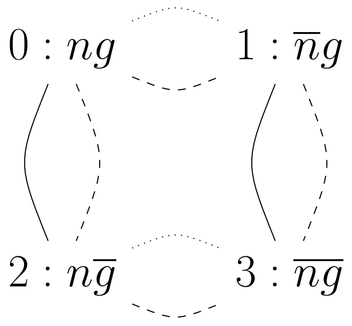
$n \overset{j}{\leftarrow} 6$

$$\begin{tikzcd} 0 : n \arrow[r, dash, dotted, bend left] \arrow[r, dash, dashed, bend right] & 1 : \overline{n} \end{tikzcd} \qquad n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$$

$g \overset{g}{\leftarrow} 5$



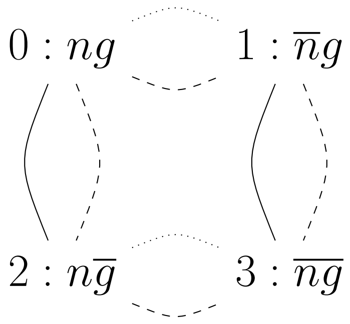
$$n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$$
$$g \mapsto 5, \overline{g} \mapsto \{M..K\} - \{5\}$$



$n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$

$g \mapsto 5, \overline{g} \mapsto \{M..K\} - \{5\}$

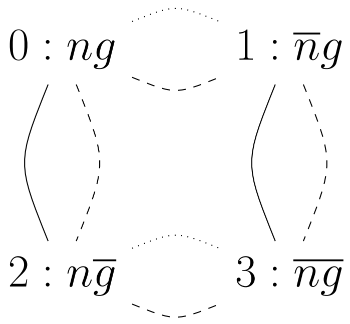
$n \neq g$



$n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$

$g \mapsto 5, \overline{g} \mapsto \{M..K\} - \{5\}$

$n \neq g$



$n \mapsto 6, \overline{n} \mapsto \{M..K\} - \{6\}$

$g \mapsto 5, \overline{g} \mapsto \{M..K\} - \{5\}$

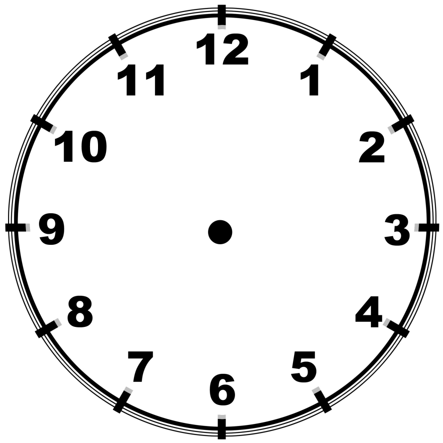
Communication: “a sends φ to b”

Sequence of commands:

$?K_a\varphi;\ \textbf{Open}\ b;\ !\varphi;\ \textbf{Close}\ b$

- $?K_a\varphi$ tests whether $K_a\varphi$ is true in the actual world.
- **Open** b adds b to the set of agents that are listening.
- $!\varphi$ is the announcement of φ.
- **Close** b removes b from the set of agents that are listening.
- As far as b is concerned, the information φ could come from anywhere: no authentication.
- Anyone who is listening receives the info: the channel is insecure.

Fast Modular Arithmetic



$$11 + 4 = 15 \equiv 3 \pmod{12}.$$

Modular division, modular inverse

Modular division is the same as multiplication by modular inverse. Modular inverses only exist for numbers that are co-prime with their modulus.

```
invM :: Integer -> Integer -> Integer
invM x n = let
   (u,v) = fct_gcd x n
   copr  = x*u + v*n == 1
   i     = if signum u == 1 then u else u+n
 in
   if copr then i else error "no inverse"
```

This uses the extended Euclidean algorithm for GCD.

The Extended Euclidean Algorithm

```
fct_gcd :: Integer -> Integer
        -> (Integer,Integer)
fct_gcd a b =
  if b == 0
  then (1,0)
  else
     let
       (q,r) = quotRem a b
       (s,t) = fct_gcd b r
     in (t, s - q*t)
```

Fast Modular Exponentiation

Modular exponentiation of x^y by repeatedly squaring modulo N.

E.g., $x^{33} \bmod 5$ can be computed by means of

$$x^{33} \bmod 5 = x^{32} \bmod 5 \times x \bmod 5.$$

$x^{32} \pmod{N}$ is computed in five steps by means of repeatedly squaring modulo N:

$$x \bmod N \rightarrow x^2 \bmod N \rightarrow x^4 \bmod N \rightarrow \ldots \rightarrow x^{32} \bmod N.$$

Fast Modular Exponentiation

Modular exponentiation of x^y by repeatedly squaring modulo N.

E.g., $x^{33} \bmod 5$ can be computed by means of

$$x^{33} \bmod 5 = x^{32} \bmod 5 \times x \bmod 5.$$

$x^{32} \pmod N$ is computed in five steps by means of repeatedly squaring modulo N:

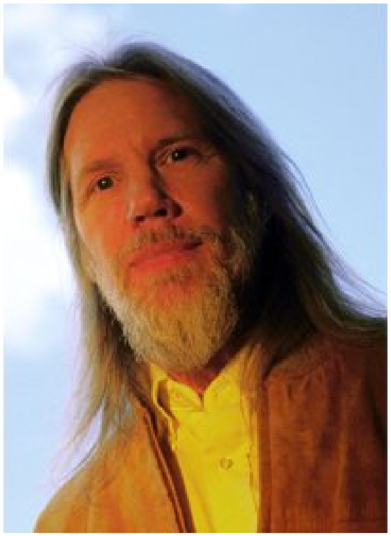
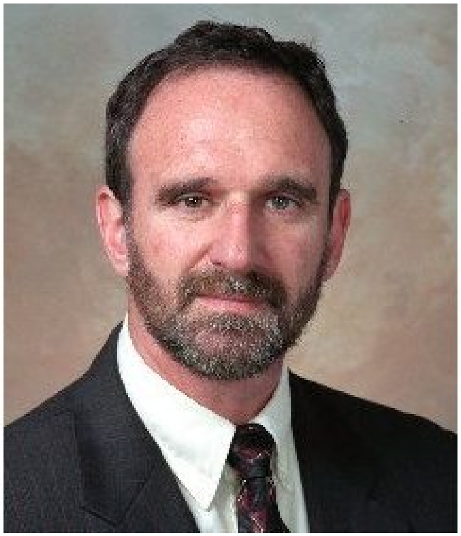
$$x \bmod N \to x^2 \bmod N \to x^4 \bmod N \to \ldots \to x^{32} \bmod N.$$

```
exM _ 0 _ = 1
exM x y n = let
   z = exM x (y `div` 2) n
   w = rem (z*z) n
 in
   if even y then w else rem (x*w) n
```

Feasible Computation

- Fast algorithms for primality testing (e.g., the probabilistic Miller-Rabin test [Mil76, Rab80]).
- Fast algorithms for co-primality testing (Euclid's GCD algorithm) and for finding modular inverses (Euclid's extended GCD algorithm, see above).
- Fast algorithms for addition and multiplication modulo, and for exponentiation modulo (see above, and compare [DK02, PP09]).
- + Articles of faith: factorisation, discrete logarithm, . . . , are not feasible.
- Now refine the meaning of "knowing a number": I know (i) the numbers that I can look up in an accessible register, and (ii) the numbers that I can feasibly compute from numbers I know.

Application: Diffie-Hellman Key Exchange [DH76]

Whitfield Diffie – Martin Hellman

Diffie-Hellman Key Exchange Protocol

Key Exchange Over Insecure Channel

1. Alice and Bob agree on a large prime p and a base $g < p$ such that g and $p - 1$ are co-prime.
2. Alice picks a secret a and sends $g^a \bmod p = A$ to Bob.
3. Bob picks a secret b and sends $g^b \bmod p = B$ to Alice.
4. Alice calculates $k = B^a \bmod p$.
5. Bob calculates $k = A^b \bmod p$.
6. They now have a shared key k. This is because $k = (g^a)^b = (g^b)^a \bmod p$.

Use of a Shared Secret Key for Secure Communication

Let p be the prime that Alice and Bob have agreed on, and let k be their shared key. Then message m is encoded as

$$m \times k \bmod p.$$

Such messages can be decoded by both Alice and Bob.

Alice knows p, k, g^b and a. She decodes cipher c with

$$c \times (g^b)^{(p-1)-a} \bmod p.$$

Bob knows p, k, g^a and b. He decodes cipher c with

$$c \times (g^a)^{(p-1)-b} \bmod p.$$

Behind this: Fermat’s Little Theorem

Pierre de Fermat (1601–1665)

If p is prime, then for every $1 \leq a < p$: $a^{p-1} \equiv 1 \pmod{p}$.

Example

```
Prelude> [ a^28 `mod` 29 | a <- [1..28] ]
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
```

Explanation

We have:

$$\begin{aligned}(g^a)^{(p-1)-b} &= g^{a((p-1)-b)} = g^{a(p-1)} \times g^{-ab} = (g^{p-1})^a \times g^{-ab} \\ &\overset{Fermat}{=} 1^a \times g^{-ab} = g^{-ab} \bmod p,\end{aligned}$$

and therefore:

$$c \times (g^a)^{(p-1)-b} \;=\; (m \times g^{ab}) \times g^{-ab} = m \times (g^{ab} \times g^{-ab}) = m \bmod p.$$

Diffie-Hellman Key Exchange as a Register Language Protocol

$a \overset{i}{\leftarrow} N$ **;** $A \overset{i}{\leftarrow} g^a \bmod p$ **;**

Open j **;** $!x = A$ **; Close** j **;**

$b \overset{j}{\leftarrow} M$ **;** $B \overset{j}{\leftarrow} g^b \bmod p$ **;**

Open i **;** $!y = B$ **; Close** i **;**

$k \overset{i}{\leftarrow} y^a \bmod p$ **;**

$k' \overset{j}{\leftarrow} x^b \bmod p$ **;**

$?k = k'$

Program for Epistemic Crypto Logic

- Formulate complete logics. A complete calculus for register logic has axioms for propositional logic, S5 axioms for the K_i operators, K axioms for the A operators, Modus Ponens, necessitation for K_i and A, and reduction axioms for the command operators, in the style of [BvEK06].
- Build an efficient model checker for this language, using register models as defined in this talk. See [Gat13] for a first prototype.
- Use the model checker for modelling cryptographic protocols, plus attacks on them.

Lesson For Formal Communication Studies

- Semantics of Natural Language $\Rightarrow$ Formal Communication Studies
- Focus on Representation of Update
- Parallels Between Natural and Programming Languages
- Haskell as Computational PTQ

```
Prelude> :t all
all :: (a -> Bool) -> [a] -> Bool
Prelude> :t any
any :: (a -> Bool) -> [a] -> Bool
```

References

[BMS98] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Bilboa, editor, Proceedings of TARK'98, pages 43–56, 1998.

[BvEK06] J. van Benthem, J. van Eijck, and B. Kooi. Logics of communication and change. Information and Computation, 204(11):1620–1662, 2006.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 22(6):644–654, 1976.

[DK02] Hans Delfs and Helmut Knebl. Introduction To Cryptography — Principles and Applications. Springer, 2002.

[ES] Jan van Eijck and François Schwarzentruber. Epistemic probability logic simplified. Under Submission.

[Fre80] Hans Freudenthal. Huygens' foundations of probability. Historia Mathematica, 7:113–117, 1980.

[Gat13] Malvin Gattinger. Epistemic crypto logic — functional programming and model checking of cryptographic protocols. Technical report, ILLC, Amsterdam, 2013. Exam paper for the course 'Functional Specification of Algorithms'.

[Hal03] J. Halpern. Reasoning About Uncertainty. MIT Press, 2003.

[Huy60] Christiaan Huygens. Van Rekeningh in Spelen van Geluck. 1660. About Calculation in Hazard Games; Latin: De ratociniis in Ludo aleae.

[Koo03] Barteld P. Kooi. Knowledge, Chance, and Change. PhD thesis, Groningen University, 2003.

[Mil76] Gary L. Miller. Riemann's hypothesis and tests for primality. Journal of Computer and System Sciences, 13(3):300–317, 1976.

[Mus89] R. Muskens. Meaning and Partiality. PhD thesis, University of Amsterdam, 1989.

[PP09] Christof Paar and Jan Pelzl. Understanding Cryptography — A Textbook for Students and Practitioners. Springer, 2009.

[Rab80] Michael O. Rabin. Probabilistic algorithm for testing primality. Journal of Number Theory, 12(1):128–138, 1980.